Proceedings of the 23rd

International Database Engineering & Applications Symposium

3

IDEAS 2019

Athens

Greece

June 10 – 12, 2019

acm

ConfSys.org

UNIVERSITÉ Concordia UNIVERSITY

HAROKOPIO UNIVERSITY

IDEAS 2019

23rd
International Database Applications & Engineering Symposium
Athens, Greece
2019-06-10 – 2019-06-12

Program Chairs

Yannis Manolopoulos, Open University of Cyprus
Mara Nikolaidou, Harokopio University of Athens

General Chairs

Bipin C. Desai, Concordia University
Dimosthenis Anagnostopoulos, Harokopio University of Athens

Local Chair

George Dimitrakopoulos, Harokopio University of Athens
Dimitris Michail, Harokopio University of Athens

Track Chairs

Béchara Albouna, University Antonine – Ticket, Lebanon
Mahmoud Barhamgi, Claude Bernard Lyon I University, France
Francesco Buccafurri, Mediterranea University of Reggio Calabria, IT
Richard Chbeir, University Pau & Pays Adour, France
Rui Chen, Samsung, USA
Roberto Nardone, Mediterranea University of Reggio Calabria, IT
Tongyuan Wang, TechEngine Plus Com
Min Xie ,Walmart Labs, USA

Editor

Bipin C. Desai, Concordia University

Sponsors
In co:operation with ACM
Bytepress/ConfSys.org
Concordia University, Harokopio University of Athens

The Association for Computing Machinery

2 Penn Plaza, Suite 701
New York, New York 10121-0701

ISBN: 978-1-4503-6249-8

Editorial production by: BytePress

Cover Artwork by: (C) Lettie

Table of Content

Invited Paper

Full Paper

Full Paper(Continued)

Full Paper(Continued)

Full Paper(Continued)

Full Paper(Continued)

Short Paper

Short Paper(Continued)

Ahmed Helal(American University of Beirut)
Iskander Gaba(American University of Beirut)

Poster Paper

Preface

These are the proceedings of the 23rd annual event of IDEAS. We find the challenge of holding a quality conference increasing with the larger number of conferences either underwritten by "not-for-profit "organizations with all invited papers or with papers guaranteed to be accepted. It is heartening to learn that in spite of these challenges, we received 81 papers this year with new approaches and ideas. This allows us to continue to be selective. This meeting highlights the current pre-occupation with big data, block chain, data analytics and the issue of personal data on the web platform; this is reflected in the accepted papers in these proceedings.

We would like to take this opportunity to thank the local and publicity chairs and the program committee for their help in the review process. All the submitted papers were assigned to four reviewers and we got back over 2.6 reviews on the average due to the shorter review periods. The proceedings consist of 31 full papers(38%), 8 short papers (16%) and 5 poster papers(11 %) .

We are honoured to have two excellent keynote speakers: Schahram Dustdar(Technical University of Vienna) and Foto Afrati,(National Technical University of Athens). The abstract and the invited paper, respectively, of these talks are included in the proceedings..

Acknowledgment: This conference would not have been possible without the help and effort of many people and organizations. Thanks are owed to:

- ACM (Anna Lacson, Craig Rodkin, and Barbara Ryan),
- BytePress, ConfSys.org, Concordia University (Nathalie Blair, Kunsheng Zhao, Ming Lu, Gerry Laval, and Will Knight),
- George Dimitrakopoulos, Dimitris Michail along with the support staff at Harokopio University of Athens who contributed selflessly and were involved in organizing and supporting the local events.

We appreciate their efforts and dedication.

Bipin C. Desai Dimosthenis Anagnostopoulos Mara Nikolaidou Yannis Manolopoulos

Reviewers
from the Program Committee

* Gergely Acs(INRIA, France)

* Foto N Afrati(National Technical University of Athens, Greece)

* Ana Sousa Almeida(Instituto Superior de Engenharia do Porto, Portugal)

* Toshiyuki Amagasa(Tsukuba University, Japan)

* Masayoshi Aritsugi(Kumamoto University, Japan)

* Ana Azevedo(Instituto Politecnico do Porto, Portugal)

* Gilbert Babin(HEC Montreal, Canada)

* Christopher Baker(University of New Brunswick, Saint John, Canada)

* Nick Bassiliades(Aristotle University of Thessaloniki, Greece)

* Ayse Bener(Ryerson University, Canada)

* Jorge Bernardino(Instituto Politecnico de Coimbra, Portugal)

* Roi Blanco(Yahoo!, Spain)

* Christophe Bobineau(Institut National Polytechnique de Grenoble, France)

* Francesco Buccafurri(University of Reggio Calabria, Italy)

* Dumitru Dan Burdescu(University of Craiova, Romania)

* Gregory Butler(Concordia University, Canada)

* Ismael Caballero(Universidad de Castilla La Mancha, Spain)

* Luciano Caroprese(University of Calabria, Italy)

* Richard Chbeir(Universite de Pau et des Pays de l'Adour, France)

* Rui Chen(Samsung, United States)

* Belkacem Chikhaoui(University of Sherbrooke, Canada)

* David Chiu(University of Puget Sound, United States)

* Martine Collard(Universite des Antilles, France)

* Carmela Comito(University of Calabria, Italy)

* Alfredo Cuzzocrea(University of Calabria, Italy)

* Jérôme Darmont(Université de Lyon, France)

* Gabriel David(Universidade do Porto, Portugal)

Reviewers
from the Program Committee
(Continued)

* Marcos Aurelio Domingues(Universidade Estadual de Maringa, Brazil)

* Anne Doucet(Universite Pierre et Marie Curie (Paris VI), France)

* Brett Drury(Scicrop, Brazil)

* Magdalini Eirinaki(San Jose State University, United States)

* Markus Endres(Universitat Augsburg, Germany)

* Nuno Escudeiro(Instituto Politecnico do Porto, Portugal)

* Bettina Fazzinga(Consiglio Nazionale delle Ricerche, Italy)

* Alvaro Figueira(Universidade do Porto, Portugal)

* Sergio Flesca(University of Calabria, Italy)

* Alberto Freitas(Universidade do Porto, Portugal)

* Filippo Furfaro(University of Calabria, Italy)

* Pedro Furtado(Universidade de Coimbra, Portugal)

* Benedict Gaster(University of the West of England, Bristol, United Kingdom)

* Sven Groppe(Medizinische Universitat Lubeck, Germany)

* Antonella Guzzo(University of Calabria, Italy)

* Marc Gyssens(Hasselt University, Belgium)

* Rawad Hammad(King's College London, University of London, United Kingdom)

* Irena Holubova (mlynkova)(Charles University Prague, Czech Repubilc)

* Michele Ianni(University of Calabria, Italy)

* Hisham Ihshaish(University of the West of England, Bristol, United Kingdom)

* Mirjana K Ivanovic(University of Novi Sad, Srebia and Montenegro)

* Nattiya Kanhabua(L3S Research Center, Germany)

* Junpei Kawamoto(Kyushu University, Japan)

* Will Knight(ConfSys.org, United States)

* Sotirios Kontogiannis(University of Ioannina, Greece)

* Michal Krátký(Technical University of Ostrava, Czech Repubilc)

Reviewers
from the Program Committee
(Continued)

* Gerry Laval(ConfSys.org, Canada)

* Georgios Lepouras(University of Peloponnese, Greece)

* Carson K. Leung(University of Manitoba, Canada)

* Chuan-ming Liu(National Taipei University of Technology, Taiwan)

* Alneu De Andrade Lopes(Universidade de Sao Paulo, Brazil)

* Grigorios Loukides(King's College London, University of London, United Kingdom)

* Jose Antonio F De Macedo(Universidade Federal do Ceara, Brazil)

* Bertil P. Marques(Instituto Superior de Engenharia do Porto, Portugal)

* Elio Masciari(Consiglio Nazionale delle Ricerche, Italy)

* Mirjana Mazuran(POLITECNICO DI MILANO, Italy)

* Giuseppe M. Mazzeo(Facebook, United States)

* Peter Mikulecky(University of Hradec Králové, Czech Repubilc)

* Dunja Mladenic(Jozef Stefan Institute, Slovenia)

* Noman Mohammed(University of Manitoba, Canada)

* Danilo Montesi(University of Bologna, Italy)

* Yang-sae Moon(Kangwon National University, Korea Republic)

* Kamran Munir(University of the West of England, Bristol, United Kingdom)

* Wilfred Ng(Hong Kong University of Science and Technology, Hong Kong)

* Yiu-kai Dennis Ng(Brigham Young University, United States)

* Mara Nikolaidou(Harokopio University, Greece)

* Selmin Nurcan(Universite Pantheon-Sorbonne (Paris I), France)

* Paulo Jorge Oliveira(Instituto Superior de Engenharia do Porto, Portugal)

* Oscar Pastor(Universidad Politecnica de Valencia, Spain)

* Valéria Magalhães Pequeno(Universidade Autonoma de Lisboa Luis de Camoes, Portugal)

* Jaroslav Pokorny(Charles University Prague, Czech Repubilc)

* Giuseppe Polese(University of Salerno, Italy)

Reviewers
from the Program Committee
(Continued)

* Luboš Popelínský(Masaryk University, Czech Repubilc)

* Filipe Portela(Universidade do Minho, Portugal)

* Chiara Pulice(University of Calabria, Italy)

* Dimitrios Rafailidis(Maastricht University, Netherlands)

* Venkatesh Raghavan(Pivotal Corporation, United States)

* Pedro Rangel Henriques(Universidade do Minho, Portugal)

* Peter Z. Revesz(University of Nebraska - Lincoln, United States)

* Marina Ribaudo(University of Genoa, Italy)

* Filipe Rodrigues(Universidade de Coimbra, Portugal)

* Fereidoon Sadri(University of North Carolina at Greensboro, Reviewer)

* Maribel Yasmina Santos(Universidade do Minho, Portugal)

* Marinette Savonnet(Universite de Bourgogne, France)

* Jianhua Shao(Cardiff University, United Kingdom)

* Atsuhiro Takasu(National Institute of Informatics, Japan)

* Giorgio Terracina(University of Calabria, Italy)

* Stephanie Teufel(University of Fribourg, Switzerland)

* Motomichi Toyama(Keio University, Japan)

* Giuseppe Tradigo(University of Calabria, Italy)

* Goce Trajcevski(Iowa State University of Science and Technology, United States)

* Irina Trubitsyna(University of Calabria, Italy)

* Jeffrey David Ullman(Stanford University, United States)

* Domenico Ursino(Università Politecnica delle Marche, Italy)

* Costas Vassilakis(University of Peloponnese, Greece)

* Krishnamurthy Vidyasankar(Memorial University of Newfoundland, Canada)

* Eugenio Vocaturo(University of Calabria, Italy)

* Pavel Vorobkalov(Volgograd State Technical University, Russian Federation)

Reviewers
from the Program Committee
(Continued)

* Alicja Wieczorkowska(Polish-Japanese Institute of Information Technology in Warsaw, Poland)

* Carlo Zaniolo(University of California, Los Angeles, United States)

* Ester Zumpano(University of Calabria, Italy)

External Reviewers

* Carlos Costa(Universidade do Minho, Portugal)

* Panagiotis Fouliras(University of Macedonia, Greece)

* Ioannis Mollas(Aristotle University of Thessaloniki, Greece)

* Roberto Nardone(University of Reggio Calabria, Italy)

* Jason Sawin(University of St. Thomas, St. Paul, United States)

* Chenghong Wang(Duke University, United States)

IDEAS Steering Committee

Desai, Bipin C. (Chair)	*Concordia University*
McClatchey, Richard	*University of the West of England, Bristol*
Ng, Wilfred	*Hong Kong Univ. of Science and Technology*
Pokorny, Jaroslav	*Charles University*
Toyoma , Motomichi	*Keio University*
Ullman, Jeffrey	*Stanford University*

Novel Paradigms for engineering large-scale resilient IoT Systems

Schahram Dustdar

Distributed Systems Group, TU Wien, Austria
dustdar@dsg.tuwien.ac.at

***Abstract*— This invited talk explores the research challenges in the domain of IoT from multiple angles and reflects on the urgently needed collective efforts from various research communities to collaborate on those. Our approach fundamentally challenges the current understanding of scientific, technological, and political paradigms in tackling the engineering of large-scale IoT systems. We discuss technical paradigms and research challenges in the domains of Cloud and Edge Computing as well as the requirements of people in such systems embedded in Smart Cities.**

***CCS Concepts*— • Security and privacy~Information flow control • Security and privacy~Firewalls • Security and privacy~Denial-of-service attacks**

***Keywords*— IoT device, IoT security, Data privacy, Data security**

ACM Reference format:

Schahram Dustdar. 2019. Novel Paradigms for engineering large-scale resilient IoT Systems In Proceedings of IDEAS 2019 conference (IDEAS2019). ACM, New York, NY, USA, 1 *page.* https://doi.org/10.1145/3331076.3331128

IDEAS 2019, June 10-12, 2019, Athens, Greece

The Homomorphism Property in Query Containment and Data Integration

Foto N. Afrati
National Technical University of Athens
afrati@gmail.com

ABSTRACT

We often add arithmetic to extend the expressiveness of query languages, tuple generating dependencies and data exchange mappings, and study the complexity of problems such as testing query containment and finding certain answers. When adding arithmetic comparisons, the complexity of such problems is higher than the complexity of their counterparts without them. It has been observed that we can achieve lower complexity if we restrict some of the comparisons to be closed or open semi-interval comparisons. Here, focusing on the problem of containment for conjunctive queries with arithmetic comparisons (CQAC queries, for short), we prove upper bounds on the computational complexity.

Our main methodology uses a general property of CQACs and tuple generating dependencies with arithmetic comparisons which is called the homomorphism property. When the homomorphism property holds, then the complexity of the above problems can be improved. We syntactically characterize subclasses of CQACs queries that have the homomorphism property, and we give a detailed proof that contains components that can be used to prove more results of the same kind. Similar methodology can be applied to achieve better upper bounds on the complexity of testing query containment under dependencies, finding query rewritings, and finding certain answers in data exchange. This is done by improving the complexity of the chase algorithm for tuple generating dependencies with arithmetic comparisons.

CCS CONCEPTS

• **Information systems** → **Relational database model**.

KEYWORDS

query containment, query rewriting, homomorphism

ACM Reference Format:
Foto N. Afrati. 2019. The Homomorphism Property in Query Containment and Data Integration. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3331076.3331127

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06...$15.00
https://doi.org/10.1145/3331076.3331127

1 INTRODUCTION

Homomorphisms are central in many database problems, such as query containment, finding rewritings for answering queries using views and the chase algorithm. For an example, suppose we want to check whether a conjunctive query is contained in another conjunctive query. Then it is necessary and sufficient to check whether there is a homomorphism from one query to the other [6]. For conjunctive queries, the query containment problem is NP-complete [6]. When we have constants that are numbers (e.g., they may represent prices, dates, weights, lengths, heights) then, often, we want to compare them by checking, e.g., whether two numbers are equal or whether one is greater than the other, etc. To reason about numbers we want to have a more expressive language than conjunctive queries and, thus, we add arithmetic comparisons to the definition of the query. We know that the query containment problem for conjunctive queries with arithmetic comparisons is Π_2^p-complete [10, 11, 19].

In previous literature [1–3, 20], it has been noticed that there are wide classes of CQACs for which the query containment problem remains in NP and these classes can be syntactically characterized. In this paper, we provide a detailed proof of results that have been presented earlier in extended abstract form in [3]. We present new results too.

For the proof, we need to go through a very careful technical analysis. Central to this analysis is the homomorphism property. Roughly (we will explain in technical terms shortly), for checking containment for CQACs we need to check for many homomorphisms and not just for one homomorphism and this is what usually affects the complexity. We characterize syntactically many cases where one homomorphism suffices to decide containment, which is what defines the *homomorphism property*. We will give an example in the next section to illustrate how we need multiple mappings.

We also consider domain information and extend the results about the homomorphism property. Domain information exploits the fact that, e.g., we do not compare a variable that represents time in minutes with a variable that represents weight in pounds. We can find such independent variables technically using a graph constructed from the arithmetic comparisons in the queries. Finally, we present a new result beyond the homomorphism property in Theorem 5.1.

Related work The homomorphism property for query containment was studied in [2, 10, 21]. Recent work can be found in [8], where the authors propose to extend graph functional dependencies with linear arithmetic expressions and arithmetic comparisons. They study the problems of testing satisfiability and related problems over integers (i.e., for nondense orders). A thorough study of the complexity of the problem of evaluating conjunctive queries with inequalities ($\neq$) is done in [12]. In [16] the complexity of evaluating conjunctive queries with arithmetic comparisons is investigated for acyclic queries, while query containment for acyclic conjunctive queries was investigated in [7]. Recent works [5, 18] have added arithmetic to extend the expressiveness of tuple generating dependencies and data exchange mappings, and studied the complexity of related problems.

2 PRELIMINARIES

We will state our results in detail first by referring to conjunctive queries with arithmetic comparisons and query containment. Thus we will define here these queries and later in the paper we will define what is a query rewriting using views. Then, we will extend the results about query containment to query rewriting and computing certain answers. We will discuss briefly the chase algorithm based on dependencies with arithmetic comparisons.

A *conjunctive query (CQ in short)* is a query of the form: $h(\overline{X})$:- $e_1(\overline{X}_1), \ldots, e_k(\overline{X}_k)$, where $h(\overline{X})$ and $e_i(\overline{X}_i)$ are atoms, i.e., they contain a relational symbol (h and e_i here) and a vector of variables and constants. The *head* $h(\overline{X})$ represents the results of the query, and $e_1 \ldots e_k$ represent database relations (also called base relations). The part of the conjunctive query on the right of symbol $:-$ is called the *body* of the query. Each atom in the body of a conjunctive query is said to be a *subgoal*. Every argument in the subgoal is either a variable or a constant. The variables in $\overline{X}$ are called *head* or *distinguished* variables, while the variables in $\overline{X}_i$ are called *body* variables of the query. A conjunctive query is said to be *safe* if all its distinguished variables also occur in its body. We only consider safe queries here. The result of a CQ when applied on the base relations (i.e., when applied on a database instance) is the set of atoms h such that there is an assignment of variables on both sides of the query that makes all atoms in the body of the query true.

A query Q_1 *is contained* in a query Q_2, denoted $Q_1 \sqsubseteq Q_2$, if for any database D of the base relations, the answer computed by Q_1 is a subset of the answer computed by Q_2, i.e., $Q_1(D) \subseteq Q_2(D)$. The two queries are *equivalent*, denoted $Q_1 \equiv Q_2$, if $Q_1 \sqsubseteq Q_2$ and $Q_2 \sqsubseteq Q_1$.

A *homomorphism* from a set of relational atoms to another set of relational atoms is a mapping of variables from one set to variables or constants of the other set that maps each variable to a single variable or constant and each constant to the same constant. Each atom of the former set should map to to an atom of the latter set with the same relational symbol.

A *containment mapping* from a conjunctive query Q_1 to a conjunctive query Q_2 is a homomorphism from the atoms in the body of Q_1 to the atoms in the body of Q_2 that maps the head of Q_1 to the head of Q_2. In this paper, when we consider homomorphisms between queries, they are always containment mappings, so we use the two terms interchangeably.

Chandra and Merlin [6] show that a conjunctive query Q_1 is contained in another conjunctive query Q_2 if and only if there is a containment mapping from Q_2 to Q_1.

Conjunctive queries with arithmetic comparisons (CQAC for short) are conjunctive queries that, besides the *ordinary* relational subgoals use also builtin subgoals that are arithmetic comparisons (AC for short), i.e., of the form $X\theta Y$ where θ is one of the following: $<, >, \leq, \geq, =, \neq$. Also, X is a variable and Y is either a variable or constant. If θ is either $<$ or $>$ we say that it is an open arithmetic comparison and if θ is either $\leq$ or $\geq$ we say that it is a closed AC. Moreover, the following assumptions must hold:

1) Values for the arguments in the arithmetic comparisons are chosen from an infinite, totally densely ordered set, such as the rationals or reals.

2) The arithmetic comparisons are not contradictory; that is, there exists an instantiation of the variables such that all the arithmetic comparisons are true.

3) All the comparisons are safe, i.e., each variable in the comparisons also appears in some ordinary subgoal.

We use "CQ" to represent "conjunctive query", "AC" for "arithmetic comparison", and "CQAC" for "conjunctive query with arithmetic comparisons."

- The notation we use for a CQAC query Q is $Q = Q_0 + \beta$, where Q_0 are the relational subgoals of Q and β are the arithmetic comparison subgoals of Q.

Definition 2.1. Let Q_1 and Q_2 be two conjunctive queries with arithmetic comparisons (CQACs). We want to test whether $Q_2 \sqsubseteq Q_1$. To do the testing, we first normalize each of Q_1 and Q_2 to Q_1' and Q_2', respectively. We *normalize* a CQAC query as follows:

- For each occurrence of a shared variable X in a normal (i.e., relational) subgoal, except for the first occurrence, replace the occurrence of X by a fresh variable X_i, and add $X = X_i$ to the comparisons of the query; and
- For each constant c in a normal subgoal, replace the constant by a fresh variable Z, and add $Z = c$ to the comparisons of the query.

Theorem 2.2[9, 21] below says that $Q_2 \sqsubseteq Q_1$ iff the comparisons in the normalized version Q_2' of Q_2 logically imply (denoted by "$\Rightarrow$") the disjunction of the images of the comparisons of the normalized version Q_1' of Q_1 under each containment mapping from the ordinary subgoals of Q_1' to the ordinary subgoals of Q_2'.

Theorem 2.2. *Let Q_1, Q_2 be CQACs, and $Q_1' = Q_{10}' + \beta_1', Q_2' = Q_{20}' + \beta_2'$ be the respective queries after normalization. Suppose there is at least one containment mapping from Q_{10}' to Q_{20}'. Let $\mu_1, \ldots, \mu_k$ be all the containment mappings from Q_{10}' to Q_{20}'. Then $Q_2 \sqsubseteq Q_1$ if and only if the following logical implication ϕ is true:*

$$\phi : \beta_2' \Rightarrow \mu_1(\beta_1') \vee \ldots \vee \mu_k(\beta_1').$$

(We refer to ϕ as the containment entailment *in the rest of this paper .)*

Proof. One of the directions is straightforward: If the containment entailment is true, then in any database that satisfies β_2', one of the $\mu_i(\beta_1')$ will be satisfied (because we deal with constants), and hence containment is proven.

For the "only-if" direction, suppose Q_2 is contained in Q_1, but the containment entailment is false. We assign constants to the variables that make this implication false. Then for all the containment mappings μ_i (for each of which $\mu_i(\beta_1')$ does not hold), the query containment is false, because we have found a counterexample database D. Database D is constructed by assigning the corresponding constants to the ordinary subgoals of Q_2. On this counterexample database D, Q_2 produces a tuple, but there is no μ_i that will make Q_1 produce the same tuple (because all $\mu_i(\beta_1')$ fail). We need to remember that, using the μ_i's, we can produce *all* homomorphisms from Q_1 to any database where the relational atoms of Q_2 map via a homomorphism. This is because the μ_i's were produced using the normalized version of the queries – and, hence, μ_i's were not constrained by duplication of variables or by constants (recall that, in a homomorphism, a variable is allowed to map to a single target and a constant is allowed to map on the same constant). □

Example 2.3. We will apply Theorem 2.2 to prove that Q_1 contains Q_2.

$$Q_1 \text{ :-} a(X_1, Y_1, Z_1), X_1 = Y_1, Z_1 < 5$$
$$Q_2 \text{ :-} a(X, Y, Z'), a(X', Y', Z), X \leq 5, Y \leq X, Z \leq Y,$$
$$X' = Y', Z' < 5$$

We have two containment mappings:
$\mu_1 : X_1 \to X, Y_1 \to Y, Z_1 \to Z'$
$\mu_2 : X_1 \to X', Y_1 \to Y', Z_1 \to Z$
Hence, $\mu_1(X_1) = X, \mu_1(Y_1) = Y, \mu_1(Z_1) = Z'$ and $\mu_2(X_1) = X', \mu_2(Y_1) = Y', \mu_2(Z_1) = Z$
The continament entailment is:

$X \leq 5 \wedge Y \leq X \wedge Z \leq Y \wedge X' = Y' \wedge Z' < 5 \Rightarrow$
$(\mu_1(X_1) = \mu_1(Y_1) \wedge \mu_1(Z_1) < 5) \vee$
$(\mu_2(X_1) = \mu_2(Y_1) \wedge \mu_2(Z_1) < 5)$
which is written equivalently:
$X \leq 5 \wedge Y \leq X \wedge Z \leq Y \wedge X' = Y' \wedge Z' < 5 \Rightarrow$
$(X = Y \wedge Z' < 5) \vee (X' = Y' \wedge Z < 5)$
The above logical implication is true.

2.1 Homomorphism Property (HP for short)

The homomorphism property defined below will be used to prove that for certain classes of CQAC queries, the query containment problem is in NP.

Definition 2.4. (*Homomorphism property.*) Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be two classes of CQAC queries. We say that *the homomorphism property holds* from $\mathcal{Q}_1$ to $\mathcal{Q}_2$ if for any pair of normalized queries $Q_1 \in \mathcal{Q}_1$ and $Q_2 \in \mathcal{Q}_2$ the following two statements are equivalent:
1. Q_2 is contained in Q_1.
2. There is a homomorphism μ from Q_{10} to Q_{20}, such that the following is true:

$$\beta_2 \Rightarrow \mu(\beta_1).$$

A trivial example where the homomorphism property holds is when both classes are equal to the class of CQs, i.e., conjunctive queries without arithmetic comparisons. Another trivial example is when $\mathcal{Q}_1$ is equal to the class of CQs and $\mathcal{Q}_2$ is equal to the class of CQACs.

The following theorem puts the query containment problem in NP when the homomorphism property holds.

Theorem 2.5. *Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be two classes of CQAC queries such that the homomorphism property holds from $\mathcal{Q}_1$ to $\mathcal{Q}_2$. Then checking containment of a query $Q_2 \in \mathcal{Q}_2$ in a query $Q_1 \in \mathcal{Q}_1$ can be done in nonodeterministic polynomial time.*

Proof. The witness is a mapping μ from Q_{10} to Q_{20}. We need to check that μ is a homomorphism and that

$$\beta_2 \Rightarrow \mu(\beta_1).$$

It is easy to see how to check the former in polynomial time. For the latter we check whether $\neg(\beta_2 \wedge \neg\mu(\beta_1))$ is true. To prove that this can be done in polynomial time we use the algorithm in Subsection 2.2. The algorithm finds the strongly connected components of a directed graph (this can be done in polynomial time for any directed graph) and argues on them. □

We will prove in the rest of the paper that the homomorphism property holds from $\mathcal{Q}_1$ to $\mathcal{Q}_2$ if we restrict $\mathcal{Q}_1$ to classes of conjunctive queries with comparisons which compare a variable to a constant. In particular, we define different classes of such comparisons in the following.

We use *var* and *const* to denote any variable or any constant. We define:

- *Semi-interval (SI for short)* arithmetic comparisons are the comparisons that compare a variable to a constant and do not use $\neq$, e.g., $X < 6$, $Y \geq 8$ are all SIs, while $X \neq 5$ is not an SI.
- *Left semi-interval (LSI for short)* arithmetic comparisons are SIs of the form *var* $\leq$ *const* or *var* $<$ *const*, where *var* is a variable and *const* is a constant. Symmetrically, we define *right semi-interval (RSI for short)* arithmetic comparisons to be of the form *var* $\geq$ *const* or *var* $>$ *const*. Thus, e.g., $X \leq 5$ is an LSI and $X \geq 5$ is an RSI.
- A *point inequality* arithmetic comparison use $\neq$ and compares a variable to a constant, e.g., $X \neq 6$, $Y \neq 8$ are all PIs, while $X \neq Y$ is not an PI because uses two variables (both X and Y are variables).

2.2 The algorithm to check satisfaction of a collection of ACs

We will present **algorithm AC-sat** which, on input a collection of ACs, checks whether there is a satisfying assignment, i.e., an assignment of real numbers to the variables that makes all ACs in the collection true. If there is not then we say that the conjunction of ACs is false or that the collection of ACs is *contradictory* or that the collection of ACs is *not consistent*.

We define the *induced directed graph* of a collection of ACs. The induced directed graph has nodes that are variables or constants. There is an edge labeled $\leq$ between two nodes n_1, n_2 if there is an AC in the collection which is $n_1 \leq n_2$. There is an edge labeled $<$ between two nodes n_1, n_2 if there is an AC in the collection which is $n_1 < n_2$. (We only label edges $<$ or $\leq$ since the other direction, $>$ or $\geq$ is indicated by the direction of the edge.) We treat each equation $X = Y$ as two ACs of the form $X \leq Y$ and $X \geq Y$ and we add edges accordingly. Finally we add "$<$" edges between all pairs of constants depending on their order.

Algorithm AC-sat We consider the induced directed graph of the collection of ACs and we find all strongly connected components of it. We say that an edge belongs to a strongly connected component if it joins two nodes in this strongly connected component.

The collection of ACs is contradictory if either of the following cases is true.

Case 1. There is a strongly connected component with two distinct variables belonging to it.

Case 2. There is a strongly connected component with an edge labeled $<$.

Case 3. There is a $A_1 \neq A_2$ AC such that A_1 and A_2 belong to the same strongly connected component and this component has only $\leq$ edges on it.

Lemma 2.6. *The* **algorithm AC-sat** *is a complete and sound procedure to check that a conjunction of ACs is contradictory.*

Proof. First we prove that this procedure is complete. I.e., we prove that if the procedure shows that the conjunction is not false then we can assign consistently constants to variables to make all ACs true.

Since either Case 1 nor Case 2 happens, all strongly connected components have $\leq$ labels and at most one constant. Thus, we assign to each of the elements of a strongly connected component the constant which is either a new constant or the constant of the component as follows: We collapse each strongly connected component to one node and the induced directed graph is reduced to an acyclic directed graph. We consider a topological sorting of this acyclic graph into a number of levels. We assign constants (different constants to different variables and constants that are different from the constants alsready present in the collection of ACs) following this topological sorting, so that constants in the next level are greater than the constants in the previous levels. This makes all ACs true.

Now we prove that this prodedure is sound. Whenever the procedure stops in Cases 1 and 2 then there is no assignment that satisfies all ACs in this strongly connected component because there is a cycle with either two distinct variables on it or with an edge labeled $<$. Whenever the procedure stops in Case 3, then A_1 and A_2 should be equal according to the strongly connected component they belong. Thus we cannot find an assignment that satisfies also the AC $A_1 \neq A_2$. □

A byproduct of the above proof which will be useful later is the following lemma.

Lemma 2.7. *A conjunction of ACs is false iff there is one $\neq$ AC, a_i, such that the conjunction of ACs that is created after dropping all $\neq$ ACs and keeping only a_i is also false.*

2.3 When normalization is not necessary

When we use only closed ACs, then normalization is not necessary:

Theorem 2.8. *Consider two CQAC queries, $Q_1 = Q_{10} + \beta_1$ and $Q_2 = Q_{20} + \beta_2$ over densely totally ordered domains. Suppose β_1 contains only $\leq$ and $\geq$, and each of β_1 and β_2 does not imply any "$=$" restrictions. Then $Q_2 \sqsubseteq Q_1$ if and only if*

$$\phi' : \beta_2 \Rightarrow \gamma_1(\beta_1) \vee \ldots \vee \gamma_l(\beta_1),$$

where $\gamma_1, \ldots, \gamma_l$ are all the containment mappings from Q_{10} to Q_{20}.

Proof. One of the directions is straightforward: If the containment entailment is true, then in any database that satisfies β_2, one of the $\gamma_i(\beta_1)$ will be satisfied (because we deal with constants), and hence containment is proven.

Now we prove the other direction: Suppose Q_2 is contained in Q_1, and the implication in the statement of the theorem is false. Then there is an assignment σ of values that are constants from a densely ordered domain to the variables that satisfies the left-hand side, $\sigma(\beta_2)$, but not the right-hand side of the containment entailment. This assignment σ can create a counterexample database. The critical observation is the following: Suppose, in this assignment σ, there are either two or more variables that are equal to the same constant not in the query or there is at least one variable that is equal to a constant appearing in the queries. Then there is another assignment σ' where this does not happen (i.e., all variables are assigned in σ' to distinct constants) and such that $\sigma'(\beta_2)$ is true. We will use σ' to create a counterexample database isomorphic to the relational subgoals of Q_2.

We create σ' from σ as follows: Suppose N is the number of variables in the queries. We choose a small value ϵ so that $N\epsilon$ is much smaller than any distance between the constants used in σ and also between the constants in the queries and between constants used in σ and between constants appearing in the queries. We create σ' as follows: Since β_2 is consistent and it does not imply "$=$", the variables that have the same value c_0 in σ form an acyclic graph on the induced directed graph of the ACs of β_2. We can choose any total order that is deduced from this acyclic graph and assign values/constants according to this order. In particular we assign distinct constants and each new constant is within a distance $N\epsilon$ from the original constant, c_0. E.g., if we have only two variables X and Y with value c_0 in σ, and we have in β_2 the AC $X \leq Y$, then in σ' we have $X = c_0 - \epsilon$ and $Y = c_0 + \epsilon$.

Claim If a conjunction η of closed ACs that use only constants appearing in Q_1 and Q_2 become true with the assignment σ' then they become true with the assignment σ too. I.e., if $\sigma'(\eta)$ is true then $\sigma(\eta)$ is true too.

Proof of the Claim. Observe that the relation between σ and σ' is the following by construction: The variables in η can be partitioned into subsets, pairwise disjoint and for each subset the following holds: a) the values of the variables in σ are equal to each other b) the values of the variables in σ' are all distinct but within a radius of $N\epsilon$ of each other and c) the values of the variables in σ' are within distance much greater than $N\epsilon$ from constants appearing in η except the constant in η that "belongs" in the particular subset. Hence, in $\sigma'(\eta)$, any AC $var \leq const$ (where $const$ is a constant appearing in η and var is a variable) has the same truth value as in $\sigma(\eta)$ for all variables except when they are compared to the constant that belongs to their subset. This concludes the proof of the Claim.

Now, we turn the relational subgoals of Q_2 into a database D' by assigning the values in σ' to variables and, for variables that do not appear in σ' we assign arbitrary distinct values much greater that any consant appearing in the queries. Since Q_2 is contained in Q_1, there must be a homomorphism h'_1 from the relational subgoals of Q_1 to D' such that the ACs in Q_1 are satisfied, i.e., such that $\sigma'(h'_1(\beta_1))$ is true. By construction, D' is isomorphic to the relational subgoals of Q_2. Hence h'_1 has an "isomorphic" homomorphism γ_m among the γ_i's for which $\sigma'(\gamma_m(\beta_1))$ is true.

Now we apply the Claim for $\eta = \gamma_m(\beta_1)$ and deduce that the assignment σ makes the $\gamma_m(\beta_1)$ true. Hence, we arrive at a contradiction. □

3 ANALYSING THE CONTAINMENT ENTAILMENT: CONTAINMENT IMPLICATIONS

In this section, we develop tools for the proofs we provide later. Consider the containment entailment in Theorem 2.2; we have dropped the primed versions of β_1 and β_2 but this is what we are referring to.

$$\beta_2 \Rightarrow \mu_1(\beta_1) \vee \ldots \vee \mu_k(\beta_1).$$

The right hand side of the containment entailment is a disjunction of disjuncts, where each disjunct is a conjunction of ACs. We can turn this, equivalently, to a conjunction of conjuncts, where each conjunct is a disjunction of ACs. We call each of these last conjuncts a *rhs-conjunct* (from right hand side conjunct). Now we can turn the containment entailment, equivalently, into a number of implications. n each implication, we keep the left hand side of the containment entailment the same and have the right hand side be one of the rhs-conjuncts. We call each such implication a *containment implication.* Since each logical implication $a \Rightarrow b$ can be turned, equivalently into a disjunction $\neg a \vee b$, we turn each containment implication into a disjunction of the form

$$\neg\beta_2 \vee rhsc_1 \vee rhsc_2 \vee \ldots$$

where $rhsc_1, rhsc_2, \ldots$ are the arithmetic comparisons from the particular rhs-conjunct. Finally, we turn each such disjunction into the negation of an expression E of the form

$$E = \beta_2 \wedge \neg rhsc_1 \wedge \neg rhsc_2 \wedge \ldots \quad (1)$$

It is easy to see that $E_r = \neg rhsc_1 \wedge \neg rhsc_2 \wedge \ldots$ has been created from the particular rhs-conjunct after we have negated each of the ACs of this rhs-conjunct. Thus, we call E_r a *reverse rhs-conjunct* – we may also refer to the ACs of this without using "reverse" and we assume it is understood. We illustrate on an example.

Example 3.1. We continue from Example 2.3. We repeat the queries considered:

$$Q_1 \text{ :-} a(X_1, Y_1, Z_1), X_1 = Y_1, Z_1 < 5$$

$$Q_2 \text{ :-} a(X, Y, Z'), a(X', Y', Z), X \leq 5, Y \leq X, Z \leq Y,$$

$$X' = Y', Z' < 5$$

Now we consider the containment entailment we built in Example 2.3. According to what we analyzed in this section, we can rewrite this containment entailment equivalently by transforming its right hand side into a conjunction, where each conjunct is a disjunction of ACs.

$X \leq 5 \wedge Y \leq X \wedge Z \leq Y \wedge X'{=}Y' \wedge Z' < 5 \Rightarrow (X{=}Y \vee X'{=}Y')$ $\wedge(X{=}Y \vee Z < 5) \wedge (Z' < 5 \vee X'{=}Y') \wedge (Z' < 5 \vee Z < 5)$

The above implication has 4 rhs-conjuncts, e.g., the $(X = Y \vee X' = Y')$ is one rhs-conjunct.

Now, we have 4 containment implications (one for each rhs-conjunct). We only list one of the containment implications:
$X \leq 5 \wedge Y \leq X \wedge Z \leq Y \wedge X' = Y' \wedge Z' < 5 \Rightarrow$
$(X = Y \vee X' = Y')$.

The above containment implication can be written equivalently as:
$\neg(X \leq 5 \wedge Y \leq X \wedge Z \leq Y \wedge X' = Y' \wedge Z' < 5) \vee$
$(X = Y \vee X' = Y')$. which can be written equivalently as:
$\neg\Big(X \leq 5 \wedge Y \leq X \wedge Z \leq Y \wedge X' = Y' \wedge Z' < 5 \wedge$
$\neg X = Y \wedge \neg(X' = Y')\Big)$.

Now we can replace $\neg(X = Y)$ with $X \neq Y$ and $\neg(X' = Y')$ with $X' \neq Y'$ and get, equivalently:
$\neg E =$
$\neg\Big(X \leq 5 \wedge Y \leq X \wedge Z \leq Y \wedge X' = Y' \wedge Z' < 5 \wedge$
$X \neq Y \wedge X' \neq Y'\Big)$.

E.g., $(X \neq Y \wedge X' \neq Y')$ is one of the 4 reverse rhs-conjuncts.

Now, we are ready to prove our first lemma:

Lemma 3.2. *Consider the containment entailment*

$$\beta_2 \Rightarrow \mu_1(\beta_1) \vee \ldots \vee \mu_k(\beta_1).$$

The containment entailment is equivalent to one with only one disjunct on the right hand side if and only if each containment implication has one disjunct d_i on the right hand side such that

$$\beta_2 \Rightarrow d_i$$

Proof. The one direction is obvious. For the other direction, we argue as follows. Towards contradiction, suppose each disjunct in the containment entailment has at least one AC, say a_i, such that the implication

$$\beta_2 \Rightarrow a_i$$

is not true. Then take the disjunction of all these a_i's and consider the corresponding containment implication that is formed by having this disjunction on its right hand side; call this containment implication c_i. The premises of the lemma says that, for each containment implication, we have at least one AC, d_i, on its rhs such that

$$\beta_2 \Rightarrow d_i$$

However, we assumed than in c_i all the ACs on the right hand side are not implied (each one of them) by β_2, hence contradiction. □

Theorem 3.3. *If the HP holds for the containment entailment then, for each containment implicationm there is an AC, a_i, on the rhs such that $\beta_2 \Rightarrow a_i$.*

The proof of the above theorem is a direct consequence of Lemma 3.2. It should be clear now, that we will use the above lemma in the following way: When we want to argue on the homomorphism property, we will focus on the containment implications rather than on the containment entailment.

Convention:

- We will refer to the expression E that we created above often. Remember E comes from a containment implication and this implication is true iff $\neg E$ is true.
- E is a conjunction of ACs. It contains ACs that come from the left hand side of the implication (and we will refer to them as lhs ACs) and ACs that come from the right hand side of the implication (and we will refer to them as rhs ACs).

The lemmas in the Appendix are used to set conditions for the containment implications to be true in special cases when some arithmetic comparisons are restricted to semi-interval comparisons. The intuition in their proof is that the rhs ACs are the ones that will introduce inconsistencies and it is critical to know how many rhs ACs are needed to introduce these inconsistencies. The HP holds only if one of them suffices to introduce inconsistencies.

4 CLASSES OF CQACS WHERE HP HOLDS (SI ACS)

In order to state the results clearly, we use the following definition.

Definition 4.1. An *AC-type* is one of the elements of the following set T_{AC}:

$$T_{AC} = \{var \leq var, var < var, var \leq const, var < const, const \leq var, const < var, var = var, var = const, var \neq var, var \neq const\}$$

An AC $X < Y$ is of type $var < var$ if both X and Y are variables. If X is a variable and Y is a constant then it is of typle $var < const$. The rest are defined similarly in the obvious way.

For an example, a closed LSI AC is of type $var \leq const$.

An *AC-family* is defined by a subset of T_{AC}. An AC belongs to a specific AC-family if it is of type that belongs in the family.

We present the main results about homomorphism property in three theorems; these results first appeared in [3]. The proofs in all three theorems below are direct consequence of Lemma 3.2 and Lemmas A.1 (for Theorems 4.2 and 4.3) and A.2 (for Theorem 4.5). The theorems below do not use all the potential of the lemmas in the appendix; however, it is easy to state more similar theorems if we look closer into the proofs of these lemmas. In [3], there are more refined statements of similar results; in the present paper we focus on the proof technique.

Theorem 4.2. *Suppose $\mathcal{Q}_1$ is the class of CQAC queries whose normalized version uses ACs from $\{var \leq const, var < const, var = var, var = cons\}$ and $\mathcal{Q}_2$ is the class of CQAC queries whose normalized version uses ACs from $T_{AC} - \{var \leq const\}$. Then, the homomorphism property holds from $\mathcal{Q}_1$ to $\mathcal{Q}_2$*

Theorem 4.3. *Suppose $\mathcal{Q}_1$ is the class of CQAC queries whose normalized version uses ACs from $\{var \leq const, var < const, var = var, var = cons\}$ and $\mathcal{Q}_2$ is the class of CQAC queries whose normalized version uses ACs from T_{AC}. Then, the homomorphism property holds from $\mathcal{Q}_1$ to $\mathcal{Q}_2$ under the condition that the considered queries $Q_1 \in \mathcal{Q}_1$ and $Q_2 \in \mathcal{Q}_2$ are such that they have the following property: There is no constant shared by a) an open LSI of Q_1, and b) a closed LSI of Q_2.*

We give an example to show that the conditions in the theorem are tight, in that we only use LSIs in both queries. Specifically Q_1 uses two ACs, one $var = const$ and the other $var < const$.

Example 4.4.

$$Q_1 \text{ :-} a(X, Y), X = 5, Y < 5$$

$$Q_2 \text{ :-} a(X, Y), a(Y, Z), X = 5, 5 \geq Y, Z < 5$$

or an arbitrarily long query:

$$Q_2' \text{ :-} a(X, Y_1), a(Y_1, Y_2), a(Y_2, Y_3), a(Y_3, Y_4), \ldots, a(Y_n, Z), X = 5, 5 \geq Y_1, 5 \geq Y_2, 5 \geq Y_3, \ldots, 5 \geq Y_n, Z < 5$$

Both Q_2 and Q_2' are contained in Q_1 and they both need more than one mappings to prove containment, hence the homomorphism property does not hold. Moreover Q_2' demonstrates that even if Q_1 has only one relational subgoal and two AC subgoals, there is a query Q_2' such that, in order to test containment of Q_2' to Q_1 we need arbitrarily many mappings. I.e., for Q_2' there are no $n-1$ mappings such that they suffice to prove that the containment entailment is true.

The following theorem considers containing queries with closed SI ACs.

Theorem 4.5. *Suppose $\mathcal{Q}_1$ is the class of CQAC queries whose normalized version uses ACs from $\{var \leq const, var \geq const, var = var, var = cons,\}$ and $\mathcal{Q}_2$ is the class of CQAC queries whose normalized version uses ACs from $T_{AC} - \{var > var, var > cons, cons > var\}$. Then, the homomorphism property holds from $\mathcal{Q}_1$ to $\mathcal{Q}_2$ under the condition that the considered query $Q_1 \in \mathcal{Q}_1$ has the property: There is a constant c such that all constants in RSI ACs of Q_1 are greater than c and all constants in LSI ACs of $Q_1$are less than c.*

5 BEYOND THE HOMOMORPHISM PROPERTY

There are other subclasses of CQAC for which the homomorphism property may not hold but the query containment problem is in NP. Theorem 5.1 is such a case; it extends significantly a result that was presented in [2].

5.1 A case in NP

Theorem 5.1. *Consider two conjunctive queries with arithmetic comparisons, Q_1 and Q_2 with ACs restricted as follows: All ACs are closed. The query Q_2 has any AC and the query Q_1 has one closed RSI and at least one closed LSI. Then testing containment of Q_2 to Q_1 is NP-complete.*

Proof. Notice that we do not need normalization in this case, according to Theorem 2.8. First we prove the claim:

Claim: If one closed RSI and any number of closed LSI are used in the query Q_1 then there is one disjunct in the containment entailment such that all ACs in this disjunct except one are directly implied by the ACs in Q_2.

Proof of the Claim: Suppose all disjuncts have more than one AC that is not directly implied. Since we have only one RSI in each disjunct, there is in each disjunct at least one LSI that is not directly implied. Thus, we can build a containment implication with all these LSIs on the rhs. This containment implication is not true because the algorithm **AC-sat** only uses one LSI from rhs to prove that the containment implication is true (see Lemma A.1). Using one rhs AC to prove the implication is equivalent to stating that there is an AC from the rhs which is directly implied by β_2; this is a contradiction.

Now, we will prove that we need a polynomial number of mappings. We again rewrite the containment entailment equivalently in different ways in order to argue appropriately. In particular, we use the following observation:

Observation: We can rewrite equivalently the implication $a \Rightarrow b \vee c$ as $a \wedge \neg b \Rightarrow c$.

Thus, we consider the containment entailment with $k+1$ disjuncts on the right hand side written as:

$$\beta_2 \Rightarrow D_1 \vee D_2 \cdots \vee D_k \vee (e_1 \wedge e_2 \wedge \cdots)$$

where we have written the first k disjuncts as $D_1, D_2, \ldots D_k$ and the disjunct D_{k+1} that has the property that is specified by the Claim above is written more analytically as $(e_1 \wedge e_2 \wedge \cdots)$. We assume that all $e_i, i = 2, \ldots$ are directly implied by β_2 (i.e., $\beta_2 \Rightarrow e_i, i = 2, \ldots$) but e_1 is not. Now we rewrite equivalently the containment entailment as:

$$\beta_2 \Rightarrow (D_1 \vee D_2 \cdots \vee D_k \vee e_1) \wedge$$
$$(D_1 \vee D_2 \cdots \vee D_k \vee e_2) \wedge \cdots$$

Since we have $\beta_2 \Rightarrow e_i, i = 2, \ldots$, we also have for $i = 2, \ldots$ that

$$\beta_2 \Rightarrow D_1 \vee D_2 \cdots \vee D_k \vee e_i$$

Thus the containment entailment can be equivalently written as:

$$\beta_2 \Rightarrow (D_1 \vee D_2 \cdots \vee D_k \vee e_1)$$

or as:

$$\beta_2 \wedge \neg e_1 \Rightarrow D_1 \vee D_2 \cdots \vee D_k \ (2)$$

Now, we call the above implication (2) a containment entailment although slightly abusively, since it does not come from any pair of CQAC queries. However, it has all the logical properties we used so far to rewrite equivalently the original containment entailment with $k+1$ disjuncts in the right hand side into the containment entailment in (2) with k disjuncts in the right hand side, i.e., with one disjunct less. Thus we can proceed in the same way on the containment entailment of (2) in order to write it equivalently as a containment entailment with $k-1$ disjuncts in the right hand side, and so on. However, each time we reduce one disjunct, we add a conjunct, which is an AC, on the left hand side of the arrow. We only have polynomially many different ACs that we can construct from the variables and the constants we have in the original containment entailment. Moreover, the AC that we transfer on the left hand side (e.g., the $\neg e_1$) is different than any other AC on the left hand side so far because we have assumed that it is not directly implied from the left hand side. Thus, after removing polynomially many disjuncts from the original containment entailment, we have a containment entailment, where all the ACs on the right hand side are directly implied by the conjunction on the left hand side. I.e., we have the following:

$$\beta_2 \wedge \neg e_1^1 \wedge \neg e_1^2 \cdots \Rightarrow D_m$$

where (without loss of generality) the e_1^is come from the original entailment from $D_{m+1}, D_{m+2}, \ldots, D_{k-1}, D_k$.

Thus we have proved that the containment entailment is true iff there are $D_m, D_{m+1} \ldots, D_k, D_{k+1}$ such that

$$\beta_2 \Rightarrow D_m \vee D_{m+1} \cdots \vee D_k \vee D_{k+1}$$

where $k-m$ is a positive integer which depends on the number of constants and variables in the queries as a polynomial.

Now, the certificate that will help us prove the problem in NP consists of all the mappings necessary to make the containment entailment true. We have proved that we only need polynomially many of those. We also have to argue that we can prove in polynomial time that the containment entailment is true. For this, we use the Claim and Lemma A.2. We add to the certificate a total order on the mappings $\gamma_i, i = 1, 2, \ldots$ that are given together with an AC from each $\gamma_i(\beta_1)$ which is the AC that is not directly implied from the current lhs ACs, as we argued in the proof above about the polynomial number of mappings. Thus the total order and the special AC are as explained in this proof above. Now, it is easy to check in polynomial time that an AC is directly implied by the current lhs ACs. It is also easy to check in polynomial time whether a disjunction of two ACs is implied by the lhs ACs. In summary, we prove that the containment entailment is true by a careful book-keeping, i.e., in a certain order, which is given by the total order on the mappings and the special ACs. □

6 THE HOMOMORPHISM PROPERTY EXTENDED

Many of the results about the homomorphism property carry over to other problems that are related to query containment. In order to do that, however, we need to express the homomorphism property in a more general way than relating classes of queries. As we have seen in the theorem we stated in Section 4, the CQAC classes $\mathcal{Q}_1$ and $\mathcal{Q}_2$ where described in terms of what ACs are allowed in queries that belong to each class. This proves appropriate for the problems we extend the HP to, and hence, we define the homomorphism property by giving the types of ACs allowed in each class of queries, as follows:

Definition 6.1. We say that a pair of sets of arithmetic comparison types $(\mathcal{B}_1, \mathcal{B}_2)$ *enables the homomorphism property* if the homomorphism property holds from $\mathcal{Q}_1$ to $\mathcal{Q}_2$, where $\mathcal{Q}_1$ is the class CQAC queries with ACs from $\mathcal{B}_1$, and $\mathcal{Q}_2$ is the class CQAC queries with ACs from $\mathcal{B}_2$.

Thus, Theorem 4.2 says that the pair $(\mathcal{B}_1, \mathcal{B}_2)$ enables the homomorphism property where $\mathcal{B}_1$ is $\{var \leq const, var < const, var = var, var = cons\}$ and $\mathcal{B}_2$ is $T_{AC} - \{var \leq const\}$.

7 USE DOMAIN INFORMATION

We begin with an example.

Example 7.1.

$$Q_1 : q_1(Trans) \text{ :-} a(Trans, X, Y), b(Trans, X', Y'), X' < Y',$$
$$X \neq Y$$
$$Q_2 : q_2(Trans) \text{ :-} a(Trans, X, Y), a(Trans1, Y, X),$$
$$b(Trans, X', Y'), X' < Y', X \neq Y$$

Relation b stores the transaction ID in attribute Trans, with the date of payment (attribute X) and the date of delivery (attribute Y). Relation a stores the transaction ID in attribute $Trans$, with the amount of prepayment (attribute X) and the amount of payment upon delivery (attribute Y).

Query Q_1 produces a report of the transactions' ID such that the date of payment (attribute X) is before the date of delivery (attribute Y), and the amount of prepayment (attribute X) and the amount of payment upon delivery (attribute Y) are not the same.

Query Q_2 is the same as Q_1 only that requires that there is another transaction $Trans1$, with the same amounts of prepayment and payment upon delivery in reverse.

It is obvious that the inequalities $\neq$ and $<$ do not interact because the first refers to an integer representing amount of dollars, whereas the other refers to dates. It is obvious that it does not make sense to compare amounts of money with dates.

Thus, the intuition is that a more efficient containment test may be possible in such cases. We will formalize it now.

7.1 Definitions and Formal Presentation

A *connected component* of a undirected graph is a maximal subgraph such that there is a path in the subgraph connecting any pair of nodes of the subgraph. A *bridge* of an undirected graph if an edge, which, when removed, increases the connected components of the graph. We know that there is an orientation of the edges of an undirected graph which converts it to a directed graph which is strongly connected if and only if the undirected graph has no bridges.

Let Q_1 and Q_2 be CQAC queries for which we want to test whether Q_2 is contained in Q_1. The following definition provides us with domain information:

Definition 7.2. We consider the containment entailment for testing whether Q_2 is contained in Q_1. We form the *containment entailment inequality graph* as follows: It is an undirected graph with nodes the variables and constants appearing in the containment entailment. Moreover, there is an edge between two nodes if there is an AC in the containment entailment connecting the corresponding variables or variable and constant. After removing all the bridges of this undirected graph we are left with a number of connected components that are pairwise disconnected. We call each of them a *containment component or c-component.* We call the set of nodes in each c-component a *domain.*

Definition 7.3. For each domain, we consider its c-component, and we define the *containment sub-entailment* to be the containment entailment where we have dropped all ACs that do not involve both variables/constants from this c-component. We form as many containment sub-entailments as we have c-components. Similarly, for each containment implication, we define the *containment sub-implication.*

Now we prove that the homomorphism property for each domain derives the homomorphism property in general in the following theorem.

THEOREM 7.4. *Suppose that, for each domain, the ACs in its c-component are such that in Q_1 the ACs are of type from $\mathcal{B}_1$, and in Q_2 the ACs are of type from $\mathcal{B}_2$, and that the pair of sets $(\mathcal{B}_1, \mathcal{B}_2)$ enables the homomorphism property. Then the homomorphism property holds from Q_1 to Q_2.*

PROOF. It is easy to see the following claim is true:

Claim: For each containment implication, the following is true: The containment implication is true iff there is a domain, so that the containment sub-implication (for this particular containment implication) is true for this domain.

The proof of the claim uses the fact that the c-components (which define domains) of the containment entailment inequality graph form a tree. Hence, there are no cycles that the **algorithm AC-sat** can use that involves more than one c-component.

If each domain has the HP, then for each containment sub-implication, there is a single AC on the right hand side which is implied by the left hand side. Now, we use Lemma 3.2 to conclude that there is a single disjunct in the containment entailment that is implied by the left hand side. □

8 REWRITING QUERIES USING VIEWS

View is the name we use for queries when they define pre-computed data (this is one of the uses of views, but, for the purposes of this section we do not need to be broader than that; for more, see, e.g., [4]). The problem of answering queries using views [13] via rewriting is as follows: given a query Q on a database schema and views V over the same schema, can we answer the query using only the answers to the views via a rewriting? I.e., can we find a query P using base relations only from the relations of the views such as for any database, D, we have $Q(D) = R(V(D))$? We will discuss here the problem of finding equivalent rewritings in the language of unions of CQACs for CQAC queries and views. First we define a contained rewiting and then use this definition to define equivalent rewritings. We need the following definition:

Definition 8.1. The *expansion* of a query P using views V, denoted by P^{exp}, is obtained from P by replacing all the views in P with their corresponding base relations and comparisons from their definitions. Non-distinguished variables in a view are replaced with fresh variables in P^{exp}.

THEOREM 8.2. *Given a query Q and a view set V, a query P is a* contained rewriting *of query Q using V if P uses only the views in V, and $P^{exp} \sqsubseteq Q$. Given a rewriting language $\mathcal{L}$ (e.g., conjunctive queries with comparisons), we call P an equivalent rewriting of Q using V with respect to $\mathcal{L}$ if P is in $\mathcal{L}$, and $P^{exp} \equiv Q$.*

THEOREM 8.3. *For CQ queries and views, in the language of CQs contains a number of subgoals which is at most equal to the number of subgoals in the query.*

In the presence of ACs the picture changes though. We may have arbitrarily long contained rewritings as the following example shows (it gets intuition from Example 4.4).

Example 8.4.

$$Q_1 \text{ :-} a(X,Y), X = 5, Y < 5$$

$$V_1 : v_1(X,Y) \text{ :-} a(X,Y), Y \leq 5, X = 5$$

$$V_3 : v_3(X,Y) \text{ :-} a(X,Y), Y < 5$$

$$V_2 : v_2(X,Y) \text{ :-} a(X,Y), X \leq 5, Y \leq 5$$

This is a contained rewriting of the query Q_1 using the three views:

$$R \text{ :-} v_1(X_1,X_2), v_2(X_2,X_3), v_2(X_3,X_4),$$

$$v_2(X_4,X_5), \ldots, v_2(X_{n-2},X_{n-1}), v_3(X_{n-1},X_n)$$

The intuition is the following: Because of the definitions of the views, we know: variable X_2 in R can be either equal to 5 or less that 5. X_1 is equal to 5 and if X_2 is less than 5 then we have found a mapping from the query to the expansion of R. Otherwise X_2 is equal to 5 and thus if X_3 is less than 5 we have found a mapping. Otherwise X_3 is equal to 5, etc. Up until we arrive (if we have to) at variable X_{n-1} which equal to 5 and we know that X_n is less than 5, hence we have found a mapping.

We compute the *canonical rewriting* of a query using the views as follows: First, we freeze the variables of the query to distinct constants and we compute the views on the thus created database. Then we de-freeze back the constants to their corresponding variables. The view tuples computed form the body of the canonical rewriting. Technically, computing the views on the database with the frozen variables is equivalent to finding a homomorphim from the view's subgoals to the query subgoals. Hence, we can derive the following theorem:

THEOREM 8.5. *Suppose query and views are CQs. Then, there is an equivalent rewriting in the language of CQs iff the canonical rewriting is such a rewriting.*

Theorem 8.5 can be extended to CQAC query and views only if the HP holds.

8.1 Homomorphism property for query rewriting

Definition 8.6. We say that the pair of sets of arithmetic comparison types $(\mathcal{B}_1, \mathcal{B}_2)$ *enables the homomorphism property for equivalent query rewriting* if

a) $(\mathcal{B}_1, \mathcal{B}_2)$ enables the homomorphism property.

b) The views have types of ACs from $\mathcal{B}_1$ and the query has types of ACs from $\mathcal{B}_2$.

We can define the canonical rewriting for the case there are arithmetic comparisons. We build it as in the case of CQs but we also have to satisfy the condition that the ACs in the query should imply the ACs in the views as they are mapped on variables of the query. Theorem 8.5 can be extended to CQAC queries and views when the HP holds:

THEOREM 8.7. *Consider query and views. The views have types of ACs from $\mathcal{B}_1$ and the query has types of ACs from $\mathcal{B}_2$. Suppose the pair of sets of arithmetic comparison types $(\mathcal{B}_1, \mathcal{B}_2)$ enables the homomorphism property for equivalent query rewriting.Then the following holds: If there is an equivalent rewriting in the language of CQACs, then the canonical rewriting is such a rewriting.*

We present an example.

Example 8.8. We use the three views of Example 8.4 and the query:

$$Q \text{ :-} a(X,Y), X < 5, Y < 5$$

The canonical database of Q is $\{a(X,Y), X < 5, Y < 5\}$. We compute the views on it and we construct the canonical rewriting, enhanced with ACs appropriately:

$$R_{can} \text{ :-} v_3(X,Y), v_2(X,Y), X < 5, Y < 5.$$

Notice that R_{can}^{exp} is equivalent to Q.

8.2 MCRS AND CERTAIN ANSWERS

When equivalent rewritings do not exist, then we find the next best thing which is maximally contained rewritings (MCRs). An MCR finds all "correct answers" (called *certain answers*). For CQs, i.e., for conjunctive queries without arithmetic comparisons, there is an efficient algorithm for finding an MCR in the language of unions of CQs [14, 15, 17]. However, we can use the homomorphism property to extend this algorithm for special cases of CQACs too [2]. In this case we have the following definition:

Definition 8.9. We say that the pair of sets of arithmetic comparison types $(\mathcal{B}_1, \mathcal{B}_2)$ *enables the homomorphism property for maximally contained query rewriting* if

a) $(\mathcal{B}_1, \mathcal{B}_2)$ enables the homomorphism property.

b) The query has types of ACs from $\mathcal{B}_1$ and the views have types of ACs from $\mathcal{B}_2$.

For this section and next section, see [4] and references therein for an extensive exposition and related work.

9 THE CHASE ALGORITHM

One way to view the *chase* algorithm is as generalizing the algorithm that computes the canonical rewriting. The chase algorithm considers tuple generating dependencies and equality generating dependencies. View definitions can be turned into tuple generating dependencies in a straightforward way. Thus, there is an alternative way to find the certain answers (for definitions of tuple generating dependencies and the chase algorithm see [4]). We turn the view definitions to tuple generating dependencies and apply the chase algorithm on the view instance. Then we compute the query on the result of the chase algorithm. Another problem where the chase aglrorithm is useful is when we check query containment under dependencies. However, if we add arithmetic comparisons to the tuple generating dependencies [5], then the chase algorithm does not work efficiently except in the case the homomorphism property holds for the tuple generating dependencies. We do not add details here, which can be found in [4]. However, we will explain informally on an example:

Example 9.1. Consider the views and query in Example 8.8. The views can be written as tuple generating dependencies (tgd for short) as follows:

$$V_1 : a(X,Y), Y \leq 5, X = 5 \rightarrow v_1(X,Y)$$
$$V_3 : a(X,Y), Y < 5 \rightarrow v_3(X,Y)$$
$$V_2 : a(X,Y), X \leq 5, Y \leq 5 \rightarrow v_2(X,Y)$$

The canonical database of Q is $\{a(X,Y), X < 5, Y < 5\}$. The chase algorithm applied on $\{a(X,Y), X < 5, Y < 5\}$ will work as follows. For each tgd it will check whether there is a homomorphism from its left hand side on $\{a(X,Y), X < 5, Y < 5\}$ that satsifies the ACs. If there is we add in $\{a(X,Y), X < 5, Y < 5\}$ a copy of the left hand side of the tgd – if there is not one. Thus, we end up with $\{a(X,Y), X < 5, Y < 5, v_3(X,Y), v_2(X,Y)\}$, which *satisfies* the given tgds because: for any homomorphism from the left hand side of tgd on $\{a(X,Y), X < 5, Y < 5, v_3(X,Y), v_2(X,Y)\}$ there is an extension of this homomorphism to a homomorphism from the atoms of both sides of the tgd on this instance. Now, the canonical rewriting can be formed by considering the view atoms in the result of the chase and it is the same as in Example 8.8.

The following theorem states the property of chase that makes it useful:

THEOREM 9.2. *Let Σ be a set of tgds, and D a database instance that satisfies the dependencies in Σ. Suppose K is a database instance, such that there exists a homomorphism h from K to D. Let K_Σ be the result of a successful finite chase on K with the set of dependencies Σ. Then the homomorphism h can be extended to a homomoprhism h' from K_Σ to D.*

10 CONCLUSIONS

The main novel technical contributions of this paper are a) big components of the proof technique (this technique appeared in sketch in [3]) that leads to results about the homomorphism property (i.e., Sections 2.2, 3 and the technical lemmas in the Appendix), b) the definition and formal proof of the results about domain information (this also appeared in preliminary form in [3]) and c) the result in Section 5.1. The result in Section 5.1 extends one that appears in [2] where containment is tested via a transformation to Datalog programs.

We believe we have arrived close to the boundaries for CQAC query containment problems as to whether the query containment problem has the homomorphism property. In that respect we list some open questions:

1. Lemma A.1 in the Appendix shows also in the direction that even in the case Q_1 (we check whether Q_2 is contained in Q_1) contains only LSI the problem may be Π_2^p-complete. The indication of this belief is that, in containment implications, sometimes even three ACs from the rhs are needed to prove satisfaction. Another candidate problem to be Π_2^p-complete is the following: When Q_1 contains two closed LSI and two closed RSI.

2. The proof technique we have in detail here can be used to check individual cases of problems whether they have the homomorphism property and find all such cases. E.g., we conjecture that the following case has the homomorhism property: both queries use LSI arithmetic comparisons and they both have no constants in ordinary relational subgoals.

3. Are there more cases besides the cases we presented here where query containment for classes of CQACs is in NP? Most importantly, do we need totally different proof techniques than the ones presented here?

4. We have not mentioned what happens when we have ACs of type *var* $\neq$ *const*. In the following two examples more than one mapping is needed to prove that Q_2 is contained in Q_1, therefore homomorphism property does not hold.

Example 10.1. Here we have Q_1 :-$a(X,Y), Y \neq 5$ and Q_2 :-$a(W',W), a(Y,Z), Z < W$.

Example 10.2. Here we have Q_1 :-$a(X,Y), Y \neq Y$ and Q_2 :-$a(W',W), a(Y,Z), a(U,X), X \leq Y, Z < W, W' \leq U$.

However we believe that it is not hard to prove that for the following case the homomorphism property holds: query Q_1 with $\neq$ and query Q_2 has only SI.

REFERENCES

[1] F. Afrati, R. Chirkova, M. Gergatsoulis, and V. Pavlaki. Finding equivalent rewritings in the presence of arithmetic comparisons. In *EDBT*, 2006.

[2] F. Afrati, C. Li, and P. Mitra. Answering queries using views with arithmetic comparisons. In *PODS*, 2002.

[3] F. Afrati, C. Li, and P. Mitra. On containment of conjunctive queries with arithmetic comparisons. In *EDBT*, 2004.

[4] F. N. Afrati and R. Chirkova. *Answering Queries Using Views.* Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2017.

[5] F. N. Afrati, C. Li, and V. Pavlaki. Data exchange in the presence of arithmetic comparisons. In *EDBT 2008, 11th International Conference on Extending Database Technology, Nantes, France, March 25-29, 2008, Proceedings*, pages 487–498, 2008.

[6] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. *STOC*, pages 77–90, 1977.

[7] J. Chekuri and A. Rajaraman. Conjunctive query containment revisited. In F. Afrati and P. Kolaitis, editors, *ICDT*, pages 56–70. volume 1186 of Lecture Notes in Computer Science Springer-Verlag, 1997.

[8] W. Fan, X. Liu, P. Lu, and C. Tian. Catching numeric inconsistencies in graphs. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 381–393, 2018.

[9] A. Gupta, Y. Sagiv, J. D. Ullman, and J. Widom. Constraint checking with partial information. In *PODS*, pages 45–55, 1994.

[10] A. Klug. On conjunctive queries containing inequalities. *Journal of the ACM*, 35(1):146–160, January 1988.

[11] P. G. Kolaitis, D. L. Martin, and M. N. Thakur. On the complexity of the containment problem for conjunctive queries with built-in predicates. In *PODS*, pages 197–204, 1998.

[12] P. Koutris, T. Milo, S. Roy, and D. Suciu. Answering conjunctive queries with inequalities. In *18th International Conference on Database Theory, ICDT 2015, March 23-27, 2015, Brussels, Belgium*, pages 76–93, 2015.

[13] A. Levy, A. O. Mendelzon, Y. Sagiv, and D. Srivastava. Answering queries using views. In *PODS*, pages 95–104, 1995.

[14] A. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proc. of VLDB*, pages 251–262, 1996.

[15] P. Mitra. An algorithm for answering queries efficiently using views. In *Proceedings of the Australasian Database Conference*, 2001.

[16] C. H. Papadimitriou and M. Yannakakis. On the complexity of database queries. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 12-14, 1997, Tucson, Arizona, USA*, pages 12–19, 1997.

[17] R. Pottinger and A. Levy. A scalable algorithm for answering queries using views. In *Proc. of VLDB*, 2000.

[18] B. ten Cate, P. G. Kolaitis, and W. Othman. Data exchange with arithmetic operations. In *Joint 2013 EDBT/ICDT Conferences, EDBT '13 Proceedings, Genoa, Italy, March 18-22, 2013*, pages 537–548, 2013.

[19] R. van der Meyden. The complexity of querying indefinite data about linearly ordered domains. In *PODS*, 1992.

[20] J. Wang, R. W. Topor, and M. J. Maher. Rewriting union queries using views. *Constraints*, 10(3):219–251, 2005.

[21] X. Zhang and Z. M. Ozsoyoglu. Some results on the containment and minimization of (in) equality queries. *Inf. Process. Lett.*, 1994.

A APPENDIX

A.1 Technical lemmas

The lemmas in this appendix are all of the same flavor, in that they have the same proof technique, thus, they could be stated in a single lemma with a long statement. We state them separately for clarity.

LEMMA A.1. *Consider the following implication:*

$$c_1 \wedge c_2 \wedge ... \Rightarrow d_1 \vee d_2 \vee ...$$

where the c_i's and d_i's are ACs and the conjunction of ACs $c_1 \wedge c_2 \wedge ...$ is consistent (i.e., it has a satisfying assignment from the set of real numbers). Then the following is true:

Suppose all c_is are from the AC-family T_{AC} (recall Definition 4.1), i.e., any AC. Suppose d_is are from the AC-family $\{var \leq const, var < const, var = var, const = var\}$ (i.e., besides equality, we use only LSI ACs). Then the implication is true iff one of the following happens:

(i) there is a single d_i from the rhs such that

$$c_1 \wedge c_2 \wedge ... \Rightarrow d_i$$

or

(ii) there are two ACs from the rhs, say d_i and d_j such that

$$c_1 \wedge c_2 \wedge ... \Rightarrow d_i \vee d_j$$

The case (ii) happens only if i) there is a constant shared a) by d_i, b) by one from the c_i's and c) by d_j and ii) d_i is an open AC,

or

(iii) there are three ACs from the rhs, say d_i, d_k and d_j such that

$$c_1 \wedge c_2 \wedge ... \Rightarrow d_i \vee d_k \vee d_j$$

The case (iii) happens only if there are two LSI from the rhs and two LSI from the lhs, all four sharing the same constant and, in addition, there is a rhs AC of type $var = var$.

One case where (i) happens always is when we do not have all of the following a) a rhs open LSI and a rhs closed LSI sharing the same variable and a b) a lhs equality of either type (i.e., either $var = var$ or $var = const$). Another case where (i) happens always is when the lhs ACs do not include any closed LSIs.

PROOF. *Convention:* We call the d_i's the rhs ACs (for right hand side ACs) and the c_i's the lhs ACs (for left hand side ACs).

In order to use the **algorithm AC-sat**, we write first the implication as $\neg E$ where

$$E = c_1 \wedge c_2 \wedge ... \wedge \neg d_1 \wedge \neg d_2 \wedge ...$$

We consider the induced graph of the ACs in E and we apply the algorithm to prove that E is false.

We consider the three cases of the **algorithm AC-sat**:

Case 1. Consider a strongly connected component with two distinct constants c_1 and c_2. Without loss of generality, suppose c_1 is adjacent to a rhs AC $d_i = c_1 \leq X$. From X, there is a path to a constant $c' \neq c_1$ (which is either c_2, which is $\neq c_1$ by our assumption or another one) such that c' is the first constant on this path. Now, if $c' < c_1$ then the edge from c' to c_1 forms a cycle with onle one rhs AC on it (because all rhs ACs are related to a constant, since they are SIs). If $c' > c_1$ then the edge from c_1 to c' forms a cycle that does not contain $d_i = c_1 \leq X$. Hence, we can proceed recursively until we find a cycle with only one rhs AC on it.

Case 2. Consider a strongly connected component with at least one edge (say $d_j = A_1 < A_2$) labeled by $<$. If this component has two distinct constants we argue as in case 1. Otherwise, it should have exactly one constant because the c_i ACs are not contradictory, hence at least one rhs AC should be in this component. Consider arbitrarily one of those rhs ACs, say $d_i = c_1 \leq X$ (or $d_i = c_1 < X$ whichever is the case). There is a path from X to A_1 and there is a path from A_2 to c_1; moreover these two paths do not contain any rhs AC edge because such edges are adjacent to constants (by definition) and we have assumed that there is only one constant on this strongly connected component. Hence we have created a cycle with an edge labeled by $<$ and with only one rhs AC on it.

Case 3. Consider a strongly connected component with exactly one constant c. All the rhs ACs/edges are adjacent to this constant. Let $c < X$ be such an rhs AC. For any node in this component there is a path to c and a path from X that and either path does not contain a rhs AC/edge. Hence a cycle is formed with only one rhs AC on it. Thus, if there is a $\neq$ AC between two nodes A_1 and A_2 of this strongly connected component, the set of contradictory ACs contain at most two rhs ACs (one for each A_1, A_2) and the $\neq$ AC (which is the negation of a d_i which is an $=$ AC). □

As regards the above lemma, it is convenient to define a sufficient set for the implication in the statement of the lemma: A *sufficient set* of rhs ACs is one for which its elements are sufficient to prove the implication if only those elements remain on the right hand side of the arrow of the implication. Thus the main conclusion of the lemma can be equivalently stated as:

There is a sufficient set of cardinality at most three.

Lemma A.2. *Consider the following implication:*

$$c_1 \wedge c_2 \wedge ... \Rightarrow d_1 \vee d_2 \vee ...$$

where the conjunction of ACs $c_1 \wedge c_2 \wedge ...$ is consistent (i.e., it has a satisfying assignment from the set of real numbers) and the d_i's are all closed SI (i.e., either LSI or RSI) comparisons. Then the implication is true iff one of the following happens:

(i) there is a single d_i from the rhs such that

$$c_1 \wedge c_2 \wedge ... \Rightarrow d_i$$

or

(ii) there are two ACs from the rhs from which one is LSI and one is RSI, say d_i and d_j such that

$$c_1 \wedge c_2 \wedge ... \Rightarrow d_i \vee d_j.$$

The case (ii) happens only if the following is false: There is a constant c such that all constants in RSI ACs on the right hand side are greater than c and all constants in LSI ACs on the right hand side are less than c.

Proof. We form the induced directed graph as we did in the first lemma in this appendix, and we reason on this graph further. We use the **algorithm AC-sat**. Here only Case 2 of the algorithm applies, i.e., there is a strongly connected component in the induced directed graph of the ACs with at least one rhs edge. We have two cases: Either this strongly connected component has only LSI or only RSI rhs ACs or it has of both kinds. In the first case, we argue as in the proof of Lemma A.1, only we have fewer cases since in the present lemma we only consider closed ACs.

For the second case, suppose a strongly connected component has two rhs ACs which are $X < a$ and $Y > b$ and they successive, i.e., there is a path from Y to X that uses only ACs from the left hand side of the implication. Then we consider the cycle that contains both. Then either of the following happens: a) the edge joining a and b forms a cycle which only contains $X < a$ and $Y > b$ from the rhs, and thus, we have proved our result or b) the edge joining a and b forms a cycle contains neither $X < a$ nor $Y > b$; so we proceed recursively considering now the new cycle that contains fewer rhs ACs on it. (Remember that the new cycle cannot contain only lhs ACs because we have assumed that they are consistent.) □

In a similar way we can prove the following lemma for SIs on the right hand side; this lemma is not used in this paper but it constitutes an interesting observation and it concludes the case with SI comparisons in the containing query or, as in the lemma, on the right hand side of the implication.

Lemma A.3. *Consider the following implication:*

$$c_1 \wedge c_2 \wedge ... \Rightarrow d_1 \vee d_2 \vee ...$$

where the c_i's and d_i's are ACs and the conjunction of ACs $c_1 \wedge c_2 \wedge ...$ is consistent (i.e., it has a satisfying assignment from the set of real numbers). Then the following is true:

Suppose all c_is are from the AC-family T_{AC}, i.e., any AC. Suppose c_is are from the AC-family $\{const \leq var, const < var, var \neq var, var = var, const = var, const < var\}$ (i.e., they use SI ACs). Then there is a sufficient set of cardinality at most five.

One case where it is guaranteed that there is a sufficient set of cardinality one is when there is a constant c such that all constants in RSI are greater than c and all constants in LSI are less than c.

One case where it is guaranteed that there is a sufficient set of cardinality at most two is when all the rhs are closed SI ACs.

Weight assignment on edges towards improved community detection

Dora Souliou
dsouliou@mail.ntua.gr
School of Electrical and Computer Engineering
National Technical University of Athens
Zografou, Greece

Petros Potikas
ppotik@cs.ntua.gr
School of Electrical and Computer Engineering
National Technical University of Athens
Zografou, Greece

Katerina Potika
katerina.potika@sjsu.edu
Department of Computer Science
San Jose State University
San Jose, California, USA

Aris Pagourtzis
pagour@cs.ntua.gr
School of Electrical and Computer Engineering
National Technical University of Athens
Zografou, Greece

ABSTRACT

During the last few decades the problem of community detection in social networks has become an important and challenging computational task. Consequently, a number of algorithms have been proposed in the relevant literature, some of which seem to solve the problem quite efficiently. The huge amount of data, however, forces for further improved techniques that can handle large and complicated networks. In this paper, we consider the effect of assigning weights on edges of unweighted network graphs and estimate their importance in community detection. In particular, we propose a new edge weight function and study its effect when used as a preprocessing step for community detection algorithms. Experimental results on a benchmark of random networks confirm our intuition that assigning weights on edges can play an important role in improving the performance of such algorithms.

CCS CONCEPTS

• **Networks → Network algorithms**; **Social media networks**;
• **General and reference** → *Experimentation*;

KEYWORDS

Community Detection, Social Networks, Neighborhood Overlap, Edge Betweenness, Modularity, Spanning Trees.

ACM Reference Format:
Dora Souliou, Petros Potikas, Katerina Potika, and Aris Pagourtzis. 2019. Weight assignment on edges towards improved community detection. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3331076.3331114

All authors contributed equally to this work; the order of authors' names is arbitrary.

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331114

1 INTRODUCTION

Different disciplines, such as computer science, society, economics, physics, and biology, model their complex data as networks. A network consists of nodes and edges connecting nodes, e.g., a social network consists of people (nodes) and relationships between people (edges). The rise of the Web and social media has created new challenges that require novel approaches and techniques. The problem we address here is that of assigning the most proper weights on edges of a network in order to improve the modularity value achieved by certain known community detection algorithms. More specifically we assign weights on edges as a preprocessing step for the Louvain algorithm [1] as well as for the recently introduced ST algorithm [7]. We experimentally evaluate both approaches and compare them against the existing methods on benchmark networks. In our first approach, we use the neighborhood overlap metric over the edge betweenness to assign weights on edges and then use these weights as an input to the weighted Louvain algorithm. In the second approach, we use the reciprocal fraction of the previous edge weight function and follow the minimum spanning tree-based approach of the ST algorithm [7].

Often, edges within the same community tend to have smaller edge betweenness centrality as compared to that of edges belonging to different communities. On the other hand, a small nover value of an edge indicates that its endpoints are likely to be in different communities. We therefore, use the ratio of the two quantities as a better indication of the degree of relationship between two connected nodes.

1.1 Terminology

Some useful terminology follows. For a graph $G(V, E)$, which models a network, where V is the set of nodes (users), and E is the set of edges (connections between nodes), we define the following notions and measures.

Bridge and Local Bridge: In [4] a bridge is defined as an edge between nodes A and B that if deleted will place the two endpoints A and B into two different groups, i.e., if that edge was the only way to connect A and B. A local bridge is defined as an edge between nodes A and B and if deleted it would extend the length of the path between A and B.

Edge Betweenness. Edge betweenness (eb) of an edge $e \in E$ of a graph G defines how important that edge is with respect to shortest paths that connect each pair of nodes in that graph. More specifically, eb(e) sums, for all pairs of nodes i, j, the ratio of the number of shortest paths between i and j using edge e over the total number of shortest paths between i and j. One can assume that if much of the traffic of a network passes through an edge (assuming that traffic is routed through shortest paths) then this edge is more likely to connect different communities.

Neighborhood Overlap: The neighborhood overlap (nover) of an edge (u, v) is the ratio of the number of common neighbors of both u and v to the number of nodes that are neighbors of either u or v. It is an embeddedness divided by the total number of neighbors of both nodes connected by that edge.

$$\mathsf{nover}(u, v) = \frac{\|N_u \cap N_v\|}{\|N_u \cup N_v\|} \tag{1}$$

If an edge is a local bridge then nover = 0. Hence, we can think of edges with very small nover value as being almost local bridges.

Modularity: The quality of the partition of a graph into communities usually is measured using a modularity principle as proposed by Girvan and Newman [9]. Modularity Q is a scalar value between $-1 \leq Q \leq 1$, and it measures the connectivity density of the nodes within the same community to the expected connectivity density of a graph with random edges on the same nodes. The larger the modularity score, the more appropriate is the partitioning of the nodes into communities. It is used to compare the communities obtained by different algorithms/methods. It is calculated as,

$$Q = \frac{1}{2m} \cdot \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \cdot \delta(c_i, c_j) \tag{2}$$

Where m is the number of edges, A_{ij} is the weight of edge (i, j), k_i is the degree of node i, c_i is the community that i belongs to and δ is a function that is $\delta(u, v) = 1$, if $u = v$ else 0.

1.2 Outline of the paper

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 describes how weights are assigned on edges and the effect of this assignment on existing algorithms. Section 4 presents the experimental results and section 5 concludes the paper.

2 STATE-OF-THE-ART METHODS FOR COMMUNITY DETECTION

Various approaches have been proposed in recent years to solve the community detection problem. Among them some of the most studied ones arehierarchical methods that are either divisive or agglomerative.

In the seminal paper of Girvan and Newman [6] they define the eb centrality measure and propose the GN algorithm that uses this measure. GN iteratively removes edges of higher eb centrality, thus forming connected components that correspond to communities. The main disadvantage of this algorithm is that it is computationally expensive (since it recomputes the eb values for all edges in each step) and thus not scalable. The running time in the worst case is $O(|E|^2|V|)$.

Recently, a new proposed algorithm named ST algorithm (it uses spanning tree in order to overcome the high computational cost of GN algorithm) [7], similar to this idea, is less time consuming while giving reasonable modularity score.

A two phase algorithm for weighted graphs was proposed in [1], known as the Louvain algorithm, that runs in $O(|E|)$ time. In the first phase, for each node it iteratively calculates the modularity obtained by including the node to the community of each of its neighbors, and then places this node into the community that gives the highest modularity. In the second phase, it creates a meta-graph in which communities are represented as meta-nodes and self-loops represent edges internal to the communities. The two phases are repeated on the meta-graph. This algorithm has a tendency to overlook small communities. In general, methods that use the modularity metric to optimize the community detection are known to suffer from the *resolution limit* effect [5], which refers to the fact that communities smaller than some threshold may not be discovered. Furthermore, the Louvain algorithm cannot efficiently explore the hierarchical structure of the network (if such a structure is present).

The idea that the nover score, being a measure of the similarity of two neighbor nodes, may improve existing algorithms if used in a preprocessing phase is investigated in [7] with the so called nover Louvain algorithm, which indeed performed better in several random graphs.

Yang et al. [11] use, the same as us, Lancichinetti-Fortunato-Radicchi benchmark graph to test eight state-of-the-art algorithms in order to find which is a good algorithm based on the properties of the graphs and other criteria. Some of the algorithms are: the Fastgreedy algorithm [3], which is a greedy community detection algorithm that is based on optimizing the modularity score, and the Label propagation algorithm [10], which considers each node to belong to the same community as the majority of its neighbours.

2.1 Our contribution

Our approach uses the eb centrality in conjunction with the nover metric in order to assign weights on the edges of graphs before calling the suitable community detection algorithm. The eb metric is also used by the GN algorithm; however, in contrast to the GN approach, we compute it once and use it in conjunction with the nover weights, in order to take a tree, thus considerably reducing the eb computations. The obtained results, from both modified algorithms, show that in most cases they perform better. Therefore weight values seem worth taking into further consideration.

3 ADDING EDGE WEIGHTS TO UNWEIGHTED GRAPHS

3.1 Edge weights for the Louvain algorithm

In Algorithm 1 (novel Louvain), this preprocessing is the only modification with respect to the original Louvain algorithm. Note that this increases by at most an $O(\Delta)$ factor the time complexity of the algorithm, where Δ is the maximum degree of the network. This is because we need $O(|E|\Delta)$ time for computing the nover of all edges, since computing the common neighbors of an edge can be done in $O(\Delta)$ time. Combining with the $O(|E|)$ complexity of the original

Louvain and the eb computation $O(|E||V|)$ ([2]), we get a total time complexity of $O(|E|(\Delta + V))$.

Algorithm 1 Louvain community detection with edge weights using nover and eb (novel Louvain).

```
Input: $G(V, E)$
Output: Set of communities $C$ of maximum modularity $Q$
  for each edge $e = (u, v) \in E$ do
      $\text{nover}(e) = |N_u \cap N_v| \ / \ |N_u \cup N_v|$
      compute $\text{eb}(e)$
      $w(e) \leftarrow \frac{\text{nover}(e)}{\text{eb}(e)}$
  end for
  $C \leftarrow \text{Louvain}(G, w)$
  return $C$
```

3.2 Edge weights for the ST algorithm

In Algorithm 2 (novelST), first we compute the weight of each edge, as the fraction of nover over eb. In the next step, we compute the minimum spanning tree of the graph using the weights of the first step. In the third step, we sort the edges in non-increasing order of their eb. Then, we iteratively remove one by one the edges of the spanning tree, starting from the first edge in the order. In each step of the iteration, we calculate the modularity of the communities that have been obtained and recalculate the edge betweenness in the rest of the tree. The spanning tree can be computed in an efficient way in a parallel manner. Furthermore, edge betweenness is computed in the resulted tree, adding little to the overall execution time.

4 EXPERIMENTAL RESULTS

In this section, we apply our ideas on existing methods, namely Louvain [1] and ST [7] and evaluate the given results. To accomplish this, we used synthetic networks produced by the benchmark of Lancichinetti et al [8] (LFR benchmark). For our experiments we used the python igraph library, in order to modify the existing implementations of Louvain and ST algorithm.

4.1 LFR benchmark

Our experiments use graphs with various number of nodes ranging from 500 to 3000. The value of the average node degree is equal to 10 and the maximum degree is equal to 15.

We summarize our results in Table 1 and Table 2.

The results presented in Table 1 show that in most cases the novel Louvain modularity slightly exceeds the one obtained by (standard) Louvain, and in the remaining cases the scores are quite close. Regarding ST algorithms, the novel ST algorithm constantly outperforms the ST algorithm of [7]. These results indicate that using edge weights often affects positively the quality of the results. Moreover, the number of communities given by each algorithm is not always the same. In Louvain-based algorithms only in a single case we obtain the same number of communities, while in the remaining cases the novel Louvain outputs a structure of higher granularity. Notably, the larger modularity increase is obtained when the number of communities found by novel Louvain is clearly larger than the ones found by Louvain; this matches the observation that Louvain fails to find small communities in large networks (resolution limit).

Algorithm 2 Community detection by neighborhood overlap and minimum spanning tree (novelST).

```
Input: $G(V, E)$
Output: Set of communities $C$ with maximum modularity $Q$
  for each edge $e = (u, v) \in E$ do
      compute $\text{eb}(e)$
      $\text{nover}(e) = |N_u \cap N_v| \ / \ |N_u \cup N_v|$
      $w(e) \leftarrow eb(e)/\text{nover}(e)$
  end for
  $G'(V, E') \leftarrow$ Minimum Spanning Tree$(G, w)$
  for each $e \in E'$ do
      $\text{eb}(e) \leftarrow$ calculate Edge Betweenness on $e$
  end for
  Initialize $C \leftarrow \{V\}$, $Q \leftarrow$ modularity of $C$ in $G(V, E)$      ▹ one community
  Sort all edges in $E'$ in non-increasing order of $\text{eb}(e)$
  while $E'$ is nonempty do
      Remove the edge $e$ of highest $\text{eb}(e)$ from $E'$      ▹ next edge in sorted list of edges
      $C' \leftarrow$ community structure implied by $E'$      ▹ set of components, partitioning $V$
      $Q' \leftarrow$ modularity of $C'$ in $G(V, E)$      ▹ modularity is wrt the original graph
      if $Q' > Q$ then
          $Q \leftarrow Q'$
          $C \leftarrow C'$
      end if
  end while
  return $C$
```

The results obtained by the ST algorithms are summarized in Table 2 and show a clear superiority of the novel ST algorithm in all cases. It seems that the effect of adding nover/eb edge weights can be significant for MST-based community detection methods, whereas the effect on the Louvain technique requires further investigation. Regarding the number of communities, we observe that novel ST usually finds fewer communities than the standard ST.

4.2 Real-world example

Moreover, we demonstrate the communities formed if we use the existing and the two proposed algorithms on the famous Zachary's Karate Club dataset [12] and is shown in Figure 1.

An illustration of the communities obtained by Louvain [1] and the algorithm ST in [7] on the famous Zachary's Karate Club dataset [12] is presented in Figure 2 and Figure 4. Additionally, for our novel Louvain and novelST the communities can be seen in Figure 3 and Figure 5. By the comparison of Louvain and novel Louvain, both find four communities and have similar modularity close to 0.41. The ST finds three communities and modularity 0.37, and the novelST finds two communities with a lower modularity of 0.33. Note, that the groundtruth for this dataset is two communities.

nodes	Louvain		novel Louvain	
	communities	modularity	communities	modularity
500	19	0.62	27	0.644
600	10	0.569	10	0.567
1500	33	0.656	38	0.662
2000	40	0.666	42	0.67
2500	40	0.671	58	0.668
3000	49	0.680	74	0.683

Table 1: Louvain method and Louvain method with weighted edges on LFS benchmark with average degree 10, maximum degree 15

nodes	ST		novel ST	
	communities	modularity	communities	modularity
500	7	0.484	5	0.532
600	6	0.376	6	0.463
750	12	0.496	6	0.516
900	13	0.483	7	0.55
1000	9	0.526	10	0.588

Table 2: ST method and novel ST on LFS benchmark with average degree 10, maximum edge 15

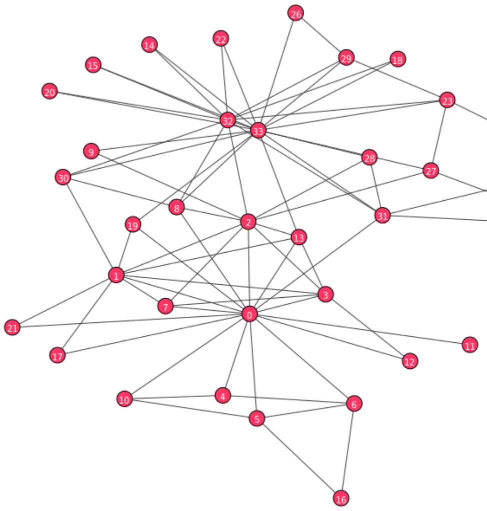


Figure 1: Zachary’s Karate Club.

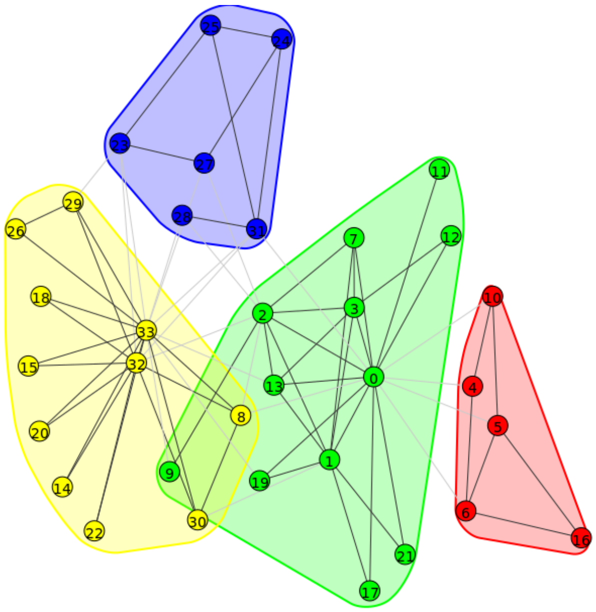


Figure 2: Zachary’s Karate Club Louvain.

5 CONCLUSION

In this paper, we propose the idea of adding edge weights on originally unweighted graphs in order to see whether we can improve known community detection algorithms. We consider two basic approaches. The first one is the well known Louvain algorithm [1] and the second is the ST algorithm [7]. From the experimental results we find out that assigning weights on edges can give increased modularity values, especially in the case of the ST algorithm. Furthermore, we notice that the structure of communities may also vary considerably. This is due not only to the weights assigned but also to the metric we use for estimating the network structure given by each algorithm, namely modularity. As a future work, we plan to investigate the importance of assigning various kinds of weights on edges in comparison to different community detection methods, possibly also using different evaluation measures.

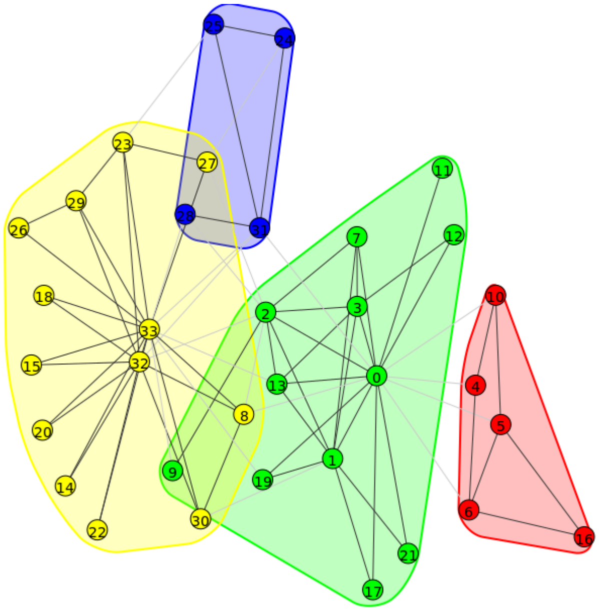


Figure 3: Zachary's Karate Club novel Louvain**.**

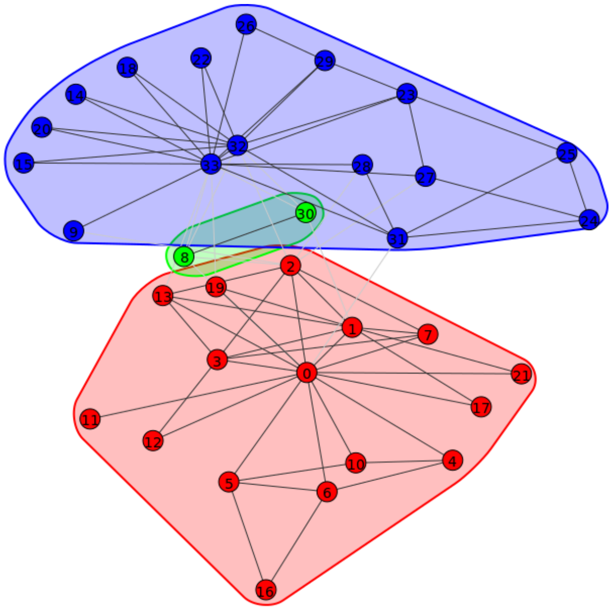


Figure 4: Zachary's Karate Club ST**.**

REFERENCES

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[2] Ulrik Brandes. A faster algorithm for betweenness centrality*. *Journal of mathematical sociology*, 25(2):163–177, 2001.

[3] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.

[4] David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world.* Cambridge University Press, 2010.

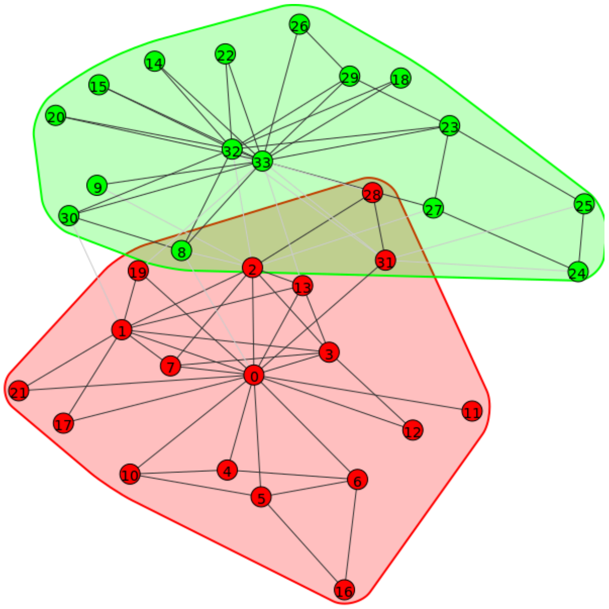


Figure 5: Zachary's Karate Club novelST.

[5] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[6] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[7] Ketki Kulkarni, Aris Pagourtzis, Katerina Potika, Petros Potikas, and Dora Souliou. Community detection via neighborhood overlap and spanning tree computations. In *Algorithmic Aspects of Cloud Computing 2018*, volume 11409 of *LNCS*, pages 13–24. Springer Nature, 2019.

[8] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78:046110, Oct 2008.

[9] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[10] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[11] Zhao Yang, René Algesheimer, and Claudio Juan Tessone. A comparative analysis of community detection algorithms on artificial networks. *CoRR*, abs/1608.00763, 2016.

[12] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, pages 452–473, 1977.

Exploratory Data Analysis and Crime Prediction for Smart Cities

Isha Pradhan
isha.pradhan@sjsu.edu
Department of Computer Science, San Jose State University
San Jose, California, USA

Katerina Potika
katerina.potika@sjsu.edu
Department of Computer Science, San Jose State University
San Jose, California, USA

Magdalini Eirinaki
magdalini.eirinaki@sjsu.edu
Department of Computer Engineering, San Jose State University
San Jose, California, USA

Petros Potikas
ppotik@cs.ntua.gr
School of Electrical and Computer Engineering, National Technical University of Athens
Zografou, Greece

ABSTRACT

Crime has been prevalent in our society for a very long time and it continues to be so even today. Currently, many cities have released crime-related data as part of an open data initiative. Using this as input, we can apply analytics to be able to predict and hopefully prevent crime in the future. In this work, we applied big data analytics to the San Francisco crime dataset, as collected by the San Francisco Police Department and available through the Open Data initiative. The main focus is to perform an in-depth analysis of the major types of crimes that occurred in the city, observe the trend over the years, and determine how various attributes contribute to specific crimes. Furthermore, we leverage the results of the exploratory data analysis to inform the data preprocessing process, prior to training various machine learning models for crime type prediction. More specifically, the model predicts the type of crime that will occur in each district of the city. We observe that the provided dataset is highly imbalanced, thus metrics used in previous research focus mainly on the majority class, disregarding the performance of the classifiers in minority classes, and propose a methodology to improve this issue. The proposed model finds applications in resource allocation of law enforcement in a Smart City.

CCS CONCEPTS

• Information systems → Data analytics; **• Computing methodologies →** *Supervised learning by classification.*

KEYWORDS

Predictive analytics, crime prediction model, multiclass classification, smart city.

ACM Reference Format:
Isha Pradhan, Katerina Potika, Magdalini Eirinaki, and Petros Potikas. 2019. Exploratory Data Analysis and Crime Prediction for Smart Cities . In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3331076.3331114

1 INTRODUCTION

The concept smart cities encompasses several initiatives that are supported by modern technology and aim at improving the lives of the people living within the city in various domains like urban development, safety, energy and so on [19]. One of the factors that determine the quality of life in a city is the crime rate therein. Although modern cities might offer a lot of technological advancements, the basic requirement of citizens' safety still remains an open problem [11].

Crime continues to be a threat to individuals and to our society and demands serious consideration if we aim at reducing the onset or the repercussions caused by it. Hundreds of crimes are recorded daily by the data officers working alongside the law enforcement authorities throughout the United States. Many cities have signed to the Open Data initiative, thereby making this crime data accessible to the general public. The intention behind this initiative is increasing the citizens' participation in decision-making and utilizing this data to uncover interesting and useful facts [7].

The city of San Francisco is one of many that have joined this Open Data initiative. The data scientists and engineers working alongside the San Francisco Police Department (SFPD) have recorded over 100, 000 crime cases in the form of police complaints they have received [6]. With the help of this historical data, many patterns can be uncovered. This can help us predict crimes that may happen in the future and thereby help the city police better safeguard the population of the city.

Motivated by the ideal scenario, where every citizen lives in a safe environment and neighborhood, we propose some methodologies as well as some initial results that might help the law enforcement of a city predict and tackle crime. We employ the crime data set reported by SFPD over a period of 15 years (2003 to 2018) and analyze them to identify the trends of crimes over the years and predict crimes that might happen in the future. Compared to previous work that has worked with the same data, our proposed data preprocessing methodology improves prediction for the highly imbalanced dataset [1, 4, 15, 23]. We should point out that, even though our proof-of-concept in this work employs the San Francisco crime dataset, a

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331114

similar approach can be followed to analyze to any city's or region's crime data, so we hope that our approach can help with crime prevention on a larger, national and international level.

1.1 Problem Formulation

The problem being tackled in this paper can be best explained in two distinct parts:

We first perform exploratory data analysis to identify crime patterns by:

- Utilizing the crime data set by the SFPD, to observe existing patterns in the crime throughout the city of San Francisco.
- Determining the classes of crimes within different areas in the city, and analyzing the spread and impact of the crime.
- Studying the crime spread in the city based on the geographical location of each crime, the possible areas of victimization on the streets, seasonal changes in the crime rate and the type, and the hourly variations in crime.

In the second part, we employ machine learning to generate a prediction methodology to identify the type of crime that can take place in the city, at several levels:

- Using the discovered patterns of crime identified during the exploratory analysis part, we inform and improve the data pre-processing process.
- Building a prediction model that treats this problem as a multiclass classification problem, by classifying new raw (unclassified) data into one of the crime categories (classes), thereby predicting the crime that can occur.
- Addressing the problem of an imbalanced dataset, by introducing additional data preprocessing tasks aiming at improving the precision and recall for all classes (including the minority classes) of our data. This improves previous research works that have been proposed on the same dataset.

For the exploratory data analysis, we employ various data analytics tools, along with Spark for initial data preprocessing, to analyze the spread of the crime in the city, and find the crime classes. For the machine learning/prediction part, in order to build a prediction model, we build upon the first part and use different types of algorithms, such as K-Nearest Neighbor, Multi-class Logistic Regression, Decision Tree, Random Forest, and Naïve Bayes.

The rest of the paper is organized as follows: in Section 2 we present an overview of the related work; our design and implementation details are presented in Section 3, while the results of our analysis and experimental evaluation are included in Section 4. We conclude with our plans for future work in Section 5.

2 RELATED WORK

Over the years, there have been a lot of studies involving the use of predictive analytics to observe patterns in crime. Some of these techniques are more complex than others and involve the use of more than one data sets. Most of the data sets used in these researches are taken from the Open Data initiative [7] supported by the government. In this section, we will study the various techniques used by different authors which will help answer questions such as: what is the role of analytics in crime prediction, what techniques are used for data preprocessing and what are the classification techniques which have proved to be most efficient.

2.1 Temporal and Spectral Analysis

A lot of research in the area of crime analysis and prediction revolves around the analysis of spatial and temporal data. The reason for this is fairly obvious as we are dealing with geographical data spread over the span of many years.

The authors of [17] have studied the fluctuation of crime throughout the year to see if there exists a pattern with seasons. In their research, they have used the crime data from three different Canadian cities, focusing on property related crimes. According to their first hypothesis, the peaks in crime during certain time intervals can be distinctly observed in the case of cities where the seasons are more distinct. Their second hypothesis is that certain types of crimes will be more frequent in certain seasons because of their nature. They were able to validate their hypothesis using Ordinary Least Squares (OLS) Regression for Vancouver and Negative Binomial Regression for Ottawa. Since their research focused on crime seasonality, a quadratic relationship in the data was predicted. Crime peaks were observed in the Summer months as compared to Winter.

In a similar study, the authors of [2] have analyzed the crime data of two US cities - Denver, CO and Los Angeles, CA and provide a comparison of the statistical analysis of the crimes in these cities. Their approach aims at finding relationships between various criminal entities as this would help in identifying crime hotspots. To increase the efficiency of prediction, various preprocessing techniques like dimensionality reduction and missing value handling were implemented. In the analysis, they compared the percentage of crime occurrence in both cities as opposed to the count of crimes. Certain common patterns were observed in both the cities such as the fact that Sunday had the lowest rate of crime in both the cities. Also, important derivations like the safest and the most notorious district were noted. Decision Tree classifier and Naive Bayes classifier were used.

L. Venturini *et al.* [22] have discovered spatio-temporal patterns in crime using spectral analysis. The goal is to observe seasonal patterns in crime and verifying if these patterns exist for all the categories of crime or if the patterns change with the type of crime. The temporal analysis thus performed highlights that the patterns not only change with the month but also with the type of crime. Hence, the authors of [22] rightly stress the fact that models built upon this data would need to account for this variation. They have used the Lomb-Scargle periodogram [18] to highlight the seasonality of the crime as it deals better with uneven or missing data. The AstroML Python package was used to achieve this. In their paper they have described in detail how every category of crime performs when the algorithm is applied to the data. Further, the authors suggest that researchers should focus on the monthly and weekly crime patterns.

2.2 Prediction using Clustering and Classification techniques

The authors of [20] have described a method to predict the type of crime which can occur based on the given location and time.

Apart from using the data from the Portland Police Bureau (PPB), they have also included data such as ethnicity of the population, census data and so on, from other public sources to increase the accuracy of their results. Further, they have made sure that the data is balanced to avoid getting skewed results. The machine learning techniques that are applied are Support Vector Machine (SVM), Random Forest, Gradient Boosting Machines, and Neural Networks [20]. Before applying the machine learning techniques to predict the category of the crime, they have applied various preprocessing techniques such as data transformation, discretization, cleaning and reduction. Due to the large volume of data, the authors have sampled the data to less than $20,000$ rows. They used two data sets to perform their experiments - one was with the demographic information used without alterations and in the second case, they used this data to predict the missing values in the original data set. In the first case, ensemble techniques like as Random Forest or Gradient Boosting worked best, while in the second case, SVM and Neural Networks showed promising results.

Since a smart city should give importance to the safety of their citizens, the authors of [11] have designed a strategy to construct a network of clusters which can assign police patrol duties, based on the informational entropy. The idea is to find patrol locations within the city, such that the entropy is maximized. The reason for the need to maximize the entropy is that the entropy, in this case, is mapped to the variation in the clusters, i.e. more entropy means more cluster coverage [11]. The dataset used for the research is the Los Angeles County *GIS* Data. The data has around 42 different crime categories. Taking the help of a domain expert, the authors have assigned weights to these crimes based on the importance of the crime. Also, the geocode for each record is taken into consideration and the records that do not have a geocode are skipped. Because the authors in [11] are trying to maximize the entropy in this case, by considering the equation $H_{c1} = -p(c_1)lnp(c_1)$. The probability $p(c1)$ is defined as the ratio of the weight of the centroid of the crime to the weight of the system, plus the ratio of the quickest path between two centroids, to the quickest path in the whole system.

The authors of [9] have taken a unique approach towards crime classification where unstructured crime reports are classified into one of the many categories of crime using textual analysis and classification. For achieving this, the data from various sources, including but not limited to the databases which store information about traffic, criminal warrants of New Jersey (NJ) and criminal records from NJ Criminal History, was combined and preprocessed. As a part of the preprocessing activity, all the stop words, punctuations, case IDs, phone numbers and so on were removed from the data. Following this, document indexing is performed on the data to convert the text into its concise representation. In order to identify the topics or specific incident types from the concise representation, the authors used Latent Semantic Analysis (LSA). Next, the similarity between these topics was identified using the Topic Modeling technique where the closer the score is to 1, the more similar it is to the topic which was followed by Text Categorization. The classification methods used in this research were Support Vector Machines (SVM), Random Forests, Neural Networks, MAXENT (Maximum Entropy Classifier), and *SLDA* (Scaled Linear Discriminant Analysis). However, the authors observed that SVM performed consistently better of them all.

2.3 Hotspot Detection

A crime hotspot is an area where the occurrence of crime is high as compared to other locations [8]. Many researchers have taken an interest in determining crime hotspots from the given dataset. The authors of [8] mainly discuss two approaches for detecting hotspots - circular and linear. The authors also discuss the fundamentals of Spatial Scan Statistics, as a useful tool for hotspot detection. The results on the Chicago crime data set are also discussed in detail using both the approaches.

3 DESIGN AND IMPLEMENTATION

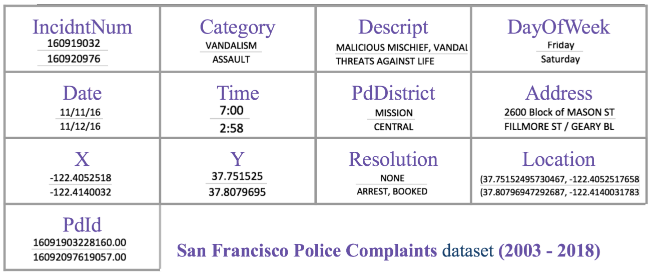
The fundamental goal of this work is to build a model, that can predict the crime category that is more likely to happen given a certain set of characteristics like the time, location, month and so on. Also, we take the help of statistical and graphical analysis to help determine which attributes contribute to the overall improvement in the Log Loss score. Our proof-of-concept application focuses on the San Francisco crime dataset. We used parallel processing using Apache Spark. Apache Spark is a big data tool which distributes the data over a cluster and achieves parallel processing. It has become popular in the recent few years [12].

3.1 Overview of the data set

We used the San Francisco crime data set [7]. The data set consists of the following attributes:

- IncidntNum: the incident number of the crime as recorded in the police logs, it is analogous to the row number,
- Descript: brief description of the crime and provides slightly more information than the *Category* field but is still quite limited,
- DayOfWeek (Date): day of the week when the crime occurred: *Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday* (exact date of the crime),
- PdDistrict: police district the crime occurred in, San Francisco has been divided into 10 police districts: *Southern, Tenderloin, Mission, Central, Northern, Bayview, Richmond, Taraval, Ingleside, Park*,
- Resolution: resolution for the crime, one of these values: *Arrested, Booked, None*,
- Address: street address of the crime,
- X (Y): longitudinal (latitudinal) coordinates of the crime,
- Location: a pair of coordinates, i.e. (X, Y),
- PdId: a unique identifier for each complaint registered for database update or search operations,
- Category: category of the crime, originally, there are 39 distinct values (such as *Assault, Larceny/Theft, Prostitution, etc.*) and it is also the dependent variable we will try to predict for the test set.

There are about 1.4 million rows and the size of it is approximately 450 MB. It contains data from the year 2003 to (February) 2018. A snapshot of the actual data set is shown in Figure 1.



IncidntNum 160919032 160920976	Category VANDALISM ASSAULT	Descript MALICIOUS MISCHIEF, VANDAL THREATS AGAINST LIFE	DayOfWeek Friday Saturday
Date 11/11/16 11/12/16	Time 7:00 2:58	PdDistrict MISSION CENTRAL	Address 2600 Block of MASON ST FILLMORE ST / GEARY BL
X -122.4052518 -122.4140032	Y 37.751525 37.8079695	Resolution NONE ARREST, BOOKED	Location (37.75152495730467, -122.4052517658 (37.80796947292687, -122.4140031783
PdId 16091903228160.00 16092097619057.00			

San Francisco Police Complaints dataset (2003 - 2018)

Figure 1: Snapshot of the actual Data set

3.2 Data Preprocessing

For our preprocessing, we employ Apache Spark. This provides several advantages, especially in terms of distributed and parallel processing. It can also significantly decrease the processing time of such a huge volume of data.

The implementation of the rest of the project has been done using Python and hence we have used the PySpark distribution of Spark for preprocessing.

The data set is mostly complete with no null values. However, there are a few outliers which must be handled (see 3.2.1). The dataset provided a lot of potential to extract more meaningful information from the existing columns. Hence, a few columns have been added or transformed to improve the score of the resulting prediction. The decision to add or transform columns has been taken by studying the graphical analysis which has been performed on the data prior to building a model.

3.2.1 Data Cleaning. One of the primary steps of data cleaning is outlier detection. Using the longitude/latitude coordinates, we identify 196 outliers that fall outside the minimum boundary of San Francisco and filter them out.

The next step in data cleaning is taking care of incorrect or missing data. Although the data set does not contain Null or missing values, the Category column does contain a few columns which have been incorrectly labeled, like the *TREA* category which should actually be *TRESPASSING*.

There are 39 distinct categories in the data set. However, some of the categories are very similar to each other. For example, when the Category column contains values or keywords like *INDECENT EXPOSURE* or *OBSCENE* or *DISORDERLY CONDUCT*, we can group those together in one category *PORNOGRAPHY/OBSCENE MAT*. The decision on which categories should be clubbed together is taken by looking at the Description column of the data set which provides more information on what the corresponding Category column represents. The complete list is presented for reference in Table 1.

3.2.2 Data Transformation. Data transformation is one of the most important data preprocessing techniques. Usually, the data is originally present in the form that makes more sense if it is transformed. In this case, the main transformations performed are as follows:

Extracting Information from Other Attributes: On taking a closer look at the Description column, it is observed that it contains a lot of useful information which has not been captured in the Category column. For example, although the Description column explains that the crime has something to do with *WEAPON LAWS*, the Category column has classified it under *OTHER OFFENSES*. This might cause us to miss out on significant information. Hence, we extract such information from the Description column and rename the categories in the Category column. The complete list is shown in Table 2 for reference.

Description Containing	New Category
License, Traffic, Speeding, Driving	Traffic Violation
Burglary Tools, Air Gun, Tear Gas, Weapon	Deadly Tool Possession
Sex	Sexual Offenses
Forgery, Fraud	Fraud/Counterfeiting
Tobacco, Drug	Drug/narcotic
Indecent Exposure, Obscene, Disorderly Conduct	Pornography/obscene Mat
Harassing	Assault
Influence Of Alcohol	Drunkenness

Table 1: Extracting Information from Description Column

Original Category containing	New Category
Weapon Laws	Deadly Tool Poss.
BadCheck, Counterfeit., Embezzl.	Fraud/Counterfeiting
Suspicious Occ	Suspicious Person/act
Warrants	Warrant Issued
Vandalism	Arson

Table 2: Combining Similar Categories

Feature Extraction: There exist several features like Address, Time, Date, X and Y which can be transformed into new features that hold more meaning as compared to the existing ones. Hence, all of these features have been used to generate new features and some of these old features have been eliminated.

Address to *BlockOrJunc*: In its original form, the Address feature has a lot of distinct values. Thus, if given a logical consideration, it is not hard to realize that the exact address of the crime might not be repeated or be useful in predicting the type of crime in the future. However, this column can be used to see if the crime occurred on a street corner/junction or on a block. We can also check if there exists a pattern among certain types of crime to occur more frequently on a street corner rather than a block. To achieve this, a simple check of whether '/' occurs in the address or not, is performed. If it does contain it, it means that the crime occurred on a corner and we return 1, otherwise it is a block and we return 0.

Time to *Hour*: The Time feature is in the Timestamp format. It would be interesting to observe patterns in crime by the hour. Hence the Hour field is extracted from the Time field. It is worth noting that if the minute part is greater than 40, i.e. if the time is for example, 12 : 42, then the hour is rounded off to 13, otherwise it would be 12.

Date to *Season, Day, Year and Month*: The Date field is a very important one for prediction. Using this single field, we are able to extract four features. Spark provides inbuilt methods to extract the Day, Month and Year from the Date and hence our script makes use of the same. After extracting the Month from the Date, we make use of this feature to extract the Season.

X and Y to *Grid*: The X and the Y coordinates provide the exact location of the crime. However, we can see some interesting patterns on dividing the entire San Francisco area into $20X20$ grids. This is inspired by the work of [15], who give specific details on the formula used for the generation of these 400 cells.

3.2.3 Data Reduction. As previously mentioned, there are 39 categories of crime in the original data set. Some of them include labels like *NON-CRIMINAL, RECOVERED VEHICLE* and *SECONDARY CODES*. Since we are trying to predict the future occurrences of crimes, it is essential to have categories pertaining to actual criminal activities. However, the above labels do not provide any additional information to help us achieve our goal. Thus, these categories are completely filtered out from our data set. This reduces the number of rows from about 2.1 million to about 1.9 million after all the preprocessing.

3.3 Classification Techniques

Classification techniques are used to automatically put the data into one or more categories also known as classes.

We focus on Pigeonhole Multiclass Classification algorithms. Multiclass Classification involves classifying the data into more than two classes. One of the most common types of Multiclass Classifiers[14] is the Pigeonhole Classifier, where every item is classified into only one of the many classes. Hence, for a given item, there can be only one output class assigned to it. Below, we briefly describe the classification techniques that we used in our analysis.

(1) Naïve Bayes classifier is a supervised learning algorithm which is based on the Bayes' theorem. The Bayes' theorem can be stated as shown in $P(A|B) = P(A)\frac{P(B|A)}{P(B)}$, where $P(A|B)$ is the conditional probability of A happening given that B is true, similar for $P(B|A)$, $P(A)$ and $P(B)$ are the individual probabilities of A and B happening independently. The Naïve Bayes classifier relaxes the conditional dependence assumption of the Bayes Theorem, introducing the "naïve" assumption that there exists independence between all pairs of features. Although these classifiers are fairly simple, they tend to work very well in a large number of real world problems.

(2) Decision Tree classifiers use decision trees to make a prediction about the value of a target variable. The decision trees are basically functions that successively determine the class that the input needs to be assigned. Using decision trees for prediction has many advantages. An input is tested against only specific subsets of the data, determined by the splitting criteria or decision functions. Another advantage is that we can use a feature selection algorithm in order to decide which features are worth considering for the decision tree classifier. The fewer the number of features, the better will the efficiency of the algorithm be [21].
To construct a decision tree, generally a top down approach is applied until some predefined stopping criterion is met.

(3) Random Forest classifiers generate multiple decision trees on different sub-samples of the data while training, and then predict the accuracy or loss score by taking a mean of these values. This helps to control over-fitting that might happen when a single decision tree is used, as this algorithm is biased towards always selecting the same root of the tree (the one that gives the less entropy after the split.
To alleviate this problem, in Random Forests the split for each node is determined from a subset of the predictor variables which are randomly chosen at the given node [16].

(4) K- Nearest Neighbor (KNN) classifiers classify data into one of the many categories by taking a majority vote of its neighbors. The label is assigned depending on the most common of the categories among its neighbors. The number of neighbors to consider is a user-defined parameter K that is set after experimentation.

(5) Multinomial Logistic Regression classifiers are a generalized version of Logistic Regression for multiclass problems like ours. The log odds of the output are modeled as a combination of the various predictor variables [5]. There are two variants of Multinomial Logistic Regression based on the nature of the distinct categories in the dependent variable-nominal and ordinal [10]. Multinomial regression uses the Maximum Likelihood Estimation (MLE) method. Logistic Regression is a discriminative classifier [13](Ch 7). This means that it tries to learn the model based on the observed data directly and makes fewer assumptions about the underlying distribution.

4 EXPERIMENTAL EVALUATION

4.1 Exploratory Data Analysis

We begin by exploring our data. Exploratory data analysis is the first step of any big data analytics process. Using graphs we can get useful and interesting insights into our data. This step will also help us make data preprocessing decisions, such as which features to include for predictions. Some of these graphs show interesting patterns in crime, which might not be apparent otherwise.

Figure 2 shows the trend of the crime over the years in various districts (neighborhoods) of San Francisco. These are the Police Districts and each of those include many other city districts. Looking at this graph, we can observe that the crime in *SOUTHERN, CENTRAL* and *NORTHERN* districts is on the rise. On the other hand, crimes in *TENDERLOIN* and *INGLESIDE* have declined over the years.

Figure 3 shows how crimes happen on different hours of the day. We can observe that there is a clear pattern in crime and the hour of the day. Generally, the crime rate is low in the early morning hours from around 3:00 AM to 6:30 AM and it rises to its peak in the evening rush hours, i.e., from 4:30 PM to 7:00 PM and is generally high at night. However, it would be really interesting to see if this pattern is followed by all the different types of crime. For this, we plot graphs for the top four crimes that we found interesting.

Figure 4, focuses on Theft/Larceny crimes per hour. It pretty much follows the trend of the previous graph.

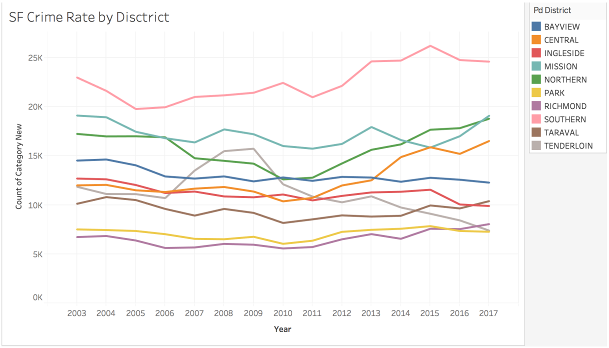


Figure 2: Rate of Crime per District by Year

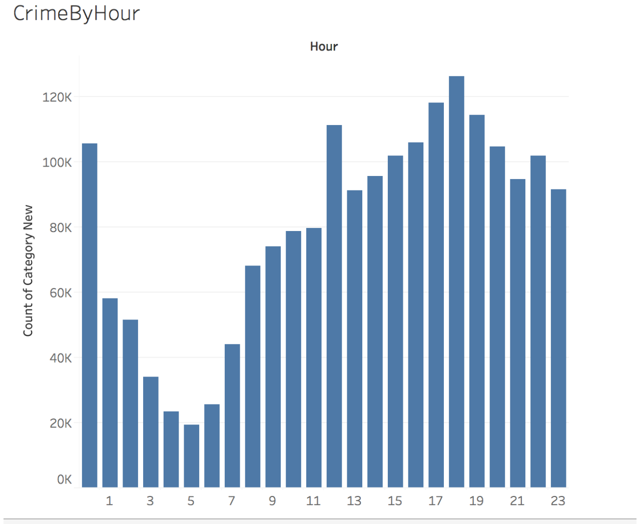


Figure 3: Rate of overall crime every Hour

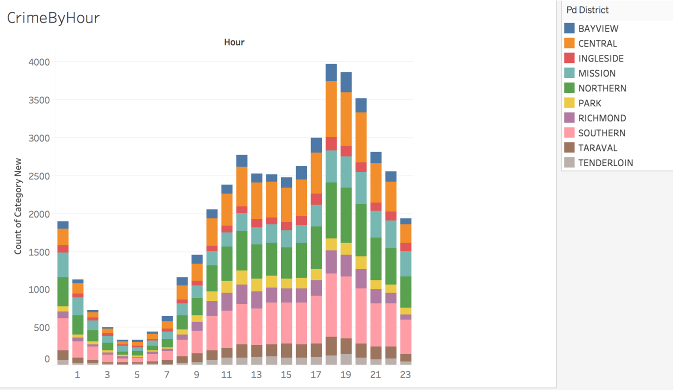


Figure 4: Rate of Theft/Larceny by the Hour

However, the same pattern is not followed by other crime types. For example, as shown in Figure 5 that plots Prostitution crimes, there are clear areas where Prostitution is high as compared to others and we can also see that Prostitution is higher during midnight and late hours (something that was expected). However, it is also very high around 11 : 00 AM in the Central district, which is unusual and can be further looked into by the police department and law enforcement agents.

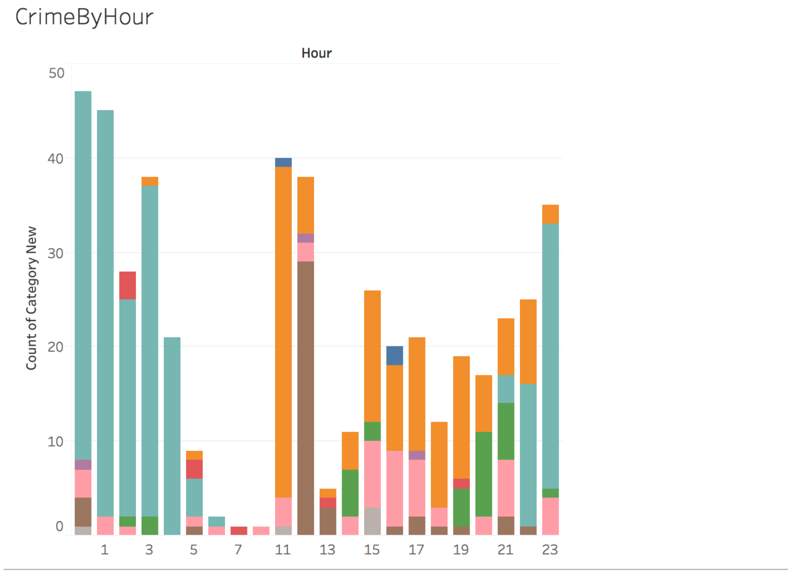


Figure 5: Rate of Prostitution by the Hour

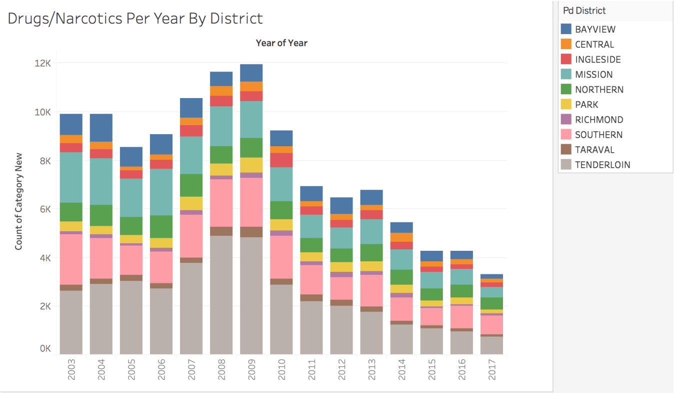


Figure 6: Sum of Drugs/Narcotics cases per Year

In Figure 6, we can see that the Drug/narcotics related crimes were highest in the year 2009 followed by 2008. Anyone even slightly familiar with San Francisco might mention that the Tenderloin is one of the most notorious districts in San Francisco with a high crime rate, especially, with high rate Drugs and Narcotics related crimes, and this impression is supported by the numbers shown here. From Figure 6 we can see that Tenderloin district has the highest number of Drug related crimes till 2009. However, in recent years, these crimes have seen a huge dip, going down by more than 50% since 2009. This might be due to the fact that

SFPD has focused their efforts on fighting crime in this notoriously crime-prone neighborhood.

A great way to study the growth or decrease in the rate of crime is by using area charts. An area chart is another way to look at the growth (or fall) rate in the data In Figure 7 we study the rise in the number of thefts over the years in most of the districts in San Francisco, except Tenderloin and Taraval. On the other hand, by plotting the area chart of Drugs and Narcotics as shown in Figure 8 we can see a clear decrease in these crimes in San Francisco.

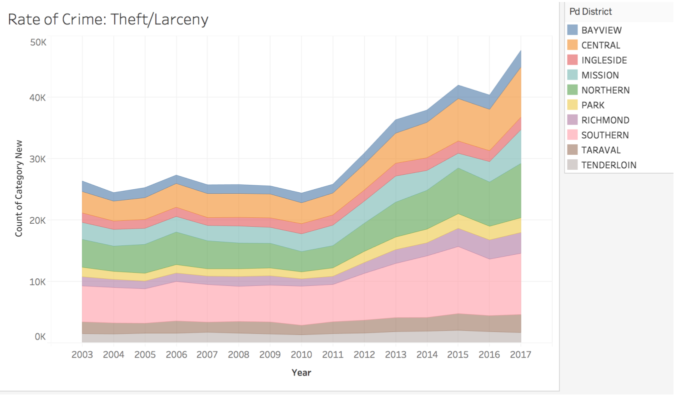


Figure 7: Area of Theft/Larceny by the Year

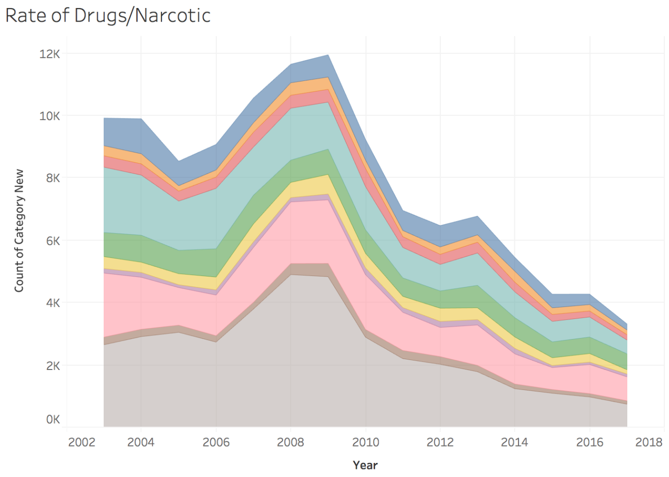


Figure 8: Area of Drugs/Narcotics by the Year

4.2 Comparison with existing results

As discussed previously, several researchers have also worked with the SF crime dataset. In this section, we provide a comparative analysis.

Our data preprocessing results in a reduction in the number of rows in the dataset from 2.19 to 1.92 million. We split the dataset to training and test chronologically as follows: as training data we use data from year 2003 to year 2015, consisting of 1, 636, 217 rows; as test data we use data from year 2016 to year 2018, consisting of 284, 165 rows.

Following the practice of researchers in related work, we use the Log Loss score for our models. In this scoring metric, false classifications are penalized. The less the Log Loss score, the better is the model. For a perfect classifier, the Log Loss score would be zero [3].

Mathematically, the Log Loss function is defined as follows:

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\, log\, p_{ij}$$

where N is the total number of samples, M is the number of distinct categories present in the output variable, y_{ij} takes the value of 0 or 1 indicating if the label j is the expected label for sample i and p_{ij} if the probability that label j will be assigned to the sample i [3].

While we cannot directly compare the results, as we cannot know how the other researchers split their datasets, we can gauge how well our approach is performing compared to this of previous work, and also provide some insights on which classifier seems to work best for this particular dataset. We first review the features used for building the various models in other existing approaches and their reported results.

Source 1 [4]: The authors have used the features *DayOfWeek, PdDistrict, X, Y, Month, Year, Hour and Grid(of 8 X 8)* for the prediction. Their best model is Random Forest with $LogLoss = 2.496$, with second best the Decision Tree with $LogLoss = 2.508$. The authors also evaluated Naive Bayes and Logistic Regression.

Source 2 [1]: The authors have used *Hour, Month, District, DayOfWeek, X, Y, Street No., Block and 3 components of PCA*. Their best model is Random Forest with $LogLoss = 2.366$, with second best the KNN classifier with $LogLoss = 2.621$. The authors also evaluated Naive Bayes.

Source 3 [23]: The attributes/features used for prediction in this work are *Year, Month, Hour, DayOfWeek, PdDistrict, X, Y and Block/Junction*. Their best model is Logistic Regression with $LogLoss = 2.45$. The authors also evaluated KNN, which yielded very high log loss.

Source 4 [15]: The features used for prediction are *Hour, DayOfWeek, Month, Year, PdDistrict, Season, BlockOrJunction, CrimeRepeatOrNot, Cell and 39-d Vector*. The authors only evaluated Logistic Regression, with $LogLoss = 2.365$

As shown in Table 3, in our approach, Random Forest is also the best model. In terms of Log Loss, our model yields the best results among the reported related work ones, with $LogLoss = 2.276$ while the second best model is the Decision tree ($LogLoss = 2.3928$).

One important aspect, left out by the previous papers focusing on crime classification in San Francisco, is the issue of data imbalance. The data set is highly skewed, as shown in the sum of the distinct categories of Figure 9. We discuss how we address this problem in what follows.

4.3 Improving Classification of Imbalanced Datasets

Most of the existing work uses accuracy or Log Loss score to evaluate the efficiency of the model. However, these metrics provide an overall assessment of the classifier, without focusing on how well

Algorithm	**Log Loss**
Random Forest	**2.2760**
Naive Bayes	2.5008
Logistic Regres.	2.4042
KNN	2.4634
Decision Tree	2.3928

Table 3: Results of Experiments (Log Loss**)**

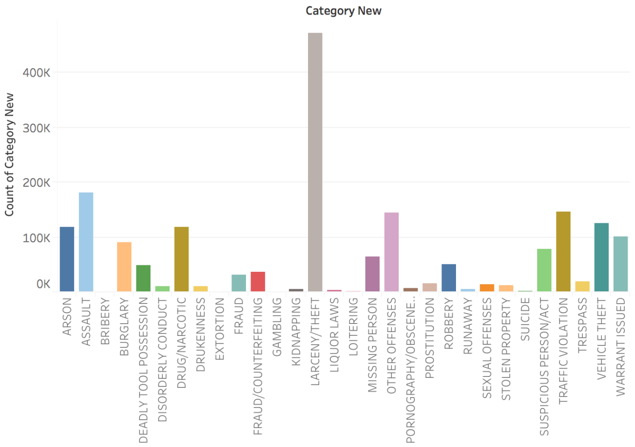


Figure 9: Count of Distinct Categories in the Dataset

the classifier does for each class. Accuracy measures the percentage of correct predictions overall predictions, so even if the classifiers don't work well with minority classes, accuracy can still be very high. The Log Loss metric does discriminate among different classes, however, it weighs each type of misclassification equally. Again, a similar misleading result might be calculated, if the classifier works well for the majority classes (which is most often the case, as it is trained using such an imbalanced dataset). Instead, we need the model to correctly identify maximum samples but at the same time, we want those correctly identified samples to include the minority classes as well, in other words increasing precision and recall for each and every class in the model.

Looking at the SF crime data, we observe that even after preprocessing, the dataset is imbalanced with the *LARCENY/THEFT* category acting as the majority class. We tried three techniques to handle the imbalance: oversampling the minority classes, oversampling the majority class, and adjusting weights on the classifiers. However, none of them showed a significant improvement in the Recall or Precision scores. Hence, the following preprocessing was performed in addition to the approaches described previously:

(1) The *LARCENY/THEFT* category was split taking into consideration the Description column. It was observed that separating out the samples with *Grand Theft From Auto* in their description proved to be a good split. The resulting classes were *LARCENY/THEFT* and *THEFT FROM AUTO.*
(2) Combined classes with less than 2000 samples into *OTHER OFFENSES* category.
(3) Created a new category called *VIOLENT/PHYSICAL CRIME* which includes former categories of *ARSON, WEAPON LAWS, VANDALISM* and instances of *ROBBERY*, where physical harm or guns were involved.

This made the dataset more balanced (see Figure 10) than the original set. We can observe this by comparing the recall of the model for the original (Figure 11) and the balanced (Figure 12) dataset.

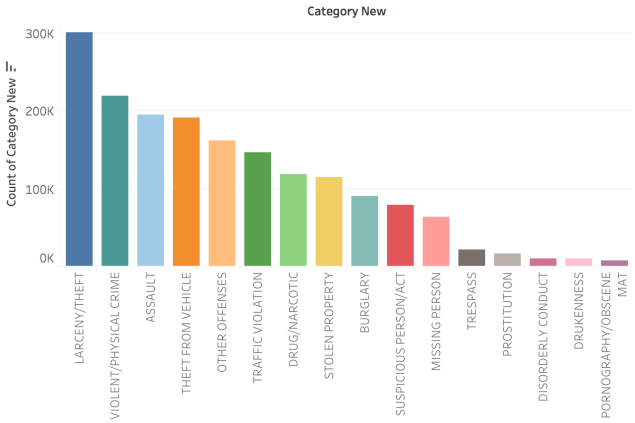


Figure 10: More Balanced data set

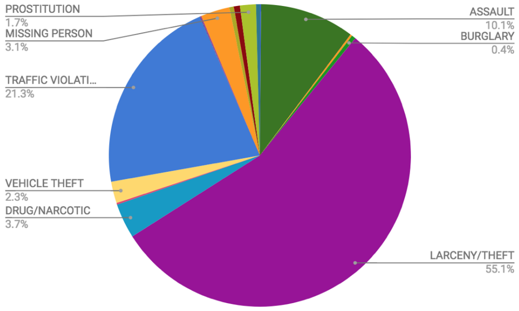


Figure 11: Recall for imbalanced dataset

5 CONCLUSION AND FUTURE WORK

In this work, we conducted a detailed analysis of the Open Data set of crime activity over 15 years for the city of San Francisco. We performed exploratory data analysis and extensive data preprocessing. Compared to previous work, we tried to alleviate the problem of an imbalanced dataset in order to improve the results of multi-class

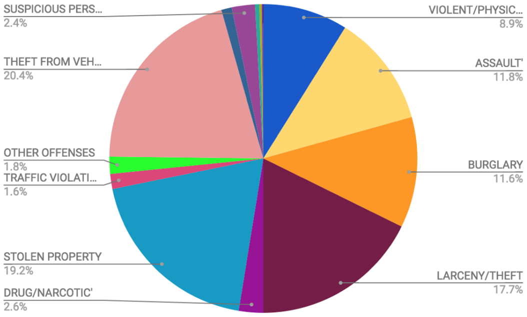


Figure 12: Recall for the more balanced dataset

classification. As a part of the future work, we plan to evaluate how other classifiers, such as neural networks, can be employed to further improve the results of the classification process. We also plan to enhance this dataset with additional metadata, such as population, housing and transportation data to gain more insights on the crime prediction process. Finally, we should stress that the proposed approach can be applied to other cities' crime datasets and see if there are any similarities and differences depending on the region.

REFERENCES

[1] Yehya Abouelnaga. San Francisco crime classification. *arXiv preprint arXiv:1607.03626*, 2016.

[2] Tahani Almanie, Rsha Mirza, and Elizabeth Lor. Crime prediction based on crime types and using spatial and temporal criminal hotspots. *arXiv preprint arXiv:1508.02050*, 2015.

[3] Exegetic Andrew B. Collier. Making Sense of Logarithmic Loss. http://www.exegetic.biz/blog/2015/12/making-sense-logarithmic-loss/, 2015.

[4] Shen Ting Ang, Weichen Wang, and Silvia Chyou. San Francisco crime classification. *University of California San Diego*, 2015.

[5] J. Bruin. Ucla: Multinomial logistic regression @ONLINE, February 2011.

[6] City and County of San Francisco. Police Department Incidents. https://data.sfgov.org/Public-Safety/Police-Department-Incidents/tmnf-yvry/, 2017.

[7] DataSF. Open government. https://www.data.gov/open-gov/. Accessed 2018-04-12.

[8] Emre Eftelioglu, Shashi Shekhar, and Xun Tang. Crime hotspot detection: A computational perspective. In *Data Mining Trends and Applications in Criminal Science and Investigations*, pages 82–111. IGI Global, 2016.

[9] Debopriya Ghosh, Soon Chun, Basit Shafiq, and Nabil R Adam. Big data-based smart city platform: Real-time crime analysis. In *Proceedings of the 17th International Digital Government Research Conference on Digital Government Research*, pages 58–66. ACM, 2016.

[10] Jelle J Goeman and Saskia le Cessie. A goodness-of-fit test for multinomial logistic regression. *Biometrics*, 62(4):980–985, 2006.

[11] Jacob Hochstetler, Lauren Hochstetler, and Song Fu. An optimal police patrol planning strategy for smart city safety. In *2016 IEEE 18th International Conference on HPCC/SmartCity/DSS*, pages 1256–1263. IEEE, 2016.

[12] Dennis Hsu, Melody Moh, and Teng-Sheng Moh. Mining frequency of drug side effects over a large twitter dataset using apache spark. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 915–924. ACM, 2017.

[13] Dan Jurafsky and James H Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2009.

[14] Brian Kolo. *Binary and Multiclass Classification*. Lulu. com, 2011.

[15] Gabriela Hernandez Larios. Case study report: San Francisco crime classification, 2016.

[16] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[17] Shannon J Linning, Martin A Andresen, and Paul J Brantingham. Crime seasonality: Examining the temporal fluctuations of property crime in cities with varying climates. *International journal of offender therapy and comparative criminology*, 61(16):1866–1891, 2017.

[18] Nicholas R Lomb. Least-squares frequency analysis of unequally spaced data. *Astrophysics and space science*, 39(2):447–462, 1976.

[19] Paolo Neirotti, Alberto De Marco, Anna Corinna Cagliano, Giulio Mangano, and Francesco Scorrano. Current trends in smart city initiatives: Some stylised facts. *Cities*, 38:25–36, 2014.

[20] Trung T Nguyen, Amartya Hatua, and Andrew H Sung. Building a learning machine classifier with inadequate data for crime prediction. *Journal of Advances in Information Technology Vol*, 8(2), 2017.

[21] Philip H Swain and Hans Hauska. The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15(3):142–147, 1977.

[22] Luca Venturini and Elena Baralis. A spectral analysis of crimes in San Francisco. In *Proceedings of the 2nd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*, page 4. ACM, 2016.

[23] Xiaoxu Wu. An informative and predictive analysis of the San Francisco police department crime data, Master Thesis, 2016.

Classification of eye-state using EEG recordings: speed-up gains using signal epochs and mutual information measure

Phoebe M Asquith
Cardiff University
Cardiff, UK
asquithpm@cardiff.ac.uk

Hisham Ihshaish
University of the West of England
Bristol, UK
hisham.ihshaish@uwe.ac.uk

ABSTRACT

The classification of electroencephalography (EEG) signals is useful in a wide range of applications such as seizure detection/prediction, motor imagery classification, emotion classification and drug effects diagnosis, amongst others. With the large number of EEG channels acquired, it has become vital that efficient data-reduction methods are developed, with varying importance from one application to another. It is also important that online classification is achieved during EEG recording for many applications, to monitor changes as they happen. In this paper we introduce a method based on Mutual Information (MI), for channel selection. Obtained results show that whilst there is a penalty on classification accuracy scores, promising speed-up gains can be achieved using MI techniques. Using MI with signal epochs (3secs) containing signal transitions enhances these speed-up gains. This work is exploratory and we suggest further research to be carried out for validation and development. Benefits to improving classification speed include improving application in clinical or educational settings.

CCS CONCEPTS

• **Mathematics of computing → Graph theory**; **Time series analysis**; • **Applied computing → Psychology**; • **Hardware → Sensor devices and platforms**.

KEYWORDS

electroencephalogram (EEG) analysis, eye-blink detection, time series analysis, graph theory applications, psychology. mutual information measure

ACM Reference Format:
Phoebe M Asquith and Hisham Ihshaish. 2019. Classification of eye-state using EEG recordings: speed-up gains using signal epochs and mutual information measure. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3331076.3331095

1 INTRODUCTION

Since its invention in 1929 [5], the electroencephalogram (EEG) has allowed the recording and interpretation of the electro-magnetic activity of neurons, from the scalp. Research using this technology has allowed crucial insights into the sleep wake cycle (e.g. [8]), neuropsychological abnormality (e.g. [17]), functional networks in the brain (e.g. [7]) and neural development (e.g. [6]).

Recently, identifying eye-state using EEG has become of interest with findings that eye-state behavior such as blink frequency can demonstrate stress response [10] or an underlying neuropsychological problem [16]. EEG signal changes related to eye-state have often been identified by separating raw data into different frequency bands [18]. However, this does not allow for online classification of eye-state.

More recently the use of portable EEGs has become more prevalent, with the development of innovative technologies (see Fig. 1). Research has demonstrated that with use of portable headsets, the eye-state of a participant can be identified using the raw time-series recorded at different channels, rather than separating data into different frequency bands [19]. Despite some concerns around the measurement capabilities of the headsets, the potential of portable devices in current and future research is recognized within the field (e.g. see [21] for review in educational research). Portable EEGs are easier to implement than traditional EEGs and can be used with subjects "in the field" or who may have difficulty sitting still (e.g. young children). Having online eye-state classification capabilities with this portable technology is an exciting step towards a dynamic resource in cognitive-neuroscientific research.

Figure 1: Example of portable EEG use with children.

The application of machine learning methods for the classification of EEG signals has been widely explored in the last two decades. Examples include methods for feature selection and optimization

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331095

as in [2] and channel selection as in [1, 22], amongst others (see [15] for a wide range of machine learning methods applied).

In previous work EEG signals have been used to classify eye-state relatively successfully using Incremental Attribute Learning (IAL) with extended timeseries [19]. Epochs of ten seconds have also been adequate for identifying drowsiness from eyestate [23]. However, to be useful as an online classifier, a shorter snapshot of data must be used to identify eye-state rather than an extended time-series, to reduce calculation time and processing power. This is also important for identifying blinks, which typically last 100-400ms [14]. Levy [13] explored the effect of epoch length on signal analysis of the EEG and found that epochs as short as 2 seconds could be used for intraoperative EEG monitoring. For eye-state classification in particular, it has been demonstrated that a snapshot of EEG signal time-series in the alpha frequency range can be used to identify eye-state, rather than an extended time-series [3].

In this research, we provide experimental analysis for sample size reduction based on a method to capture signals in discrete EEG signal slices compared to longer EEG signal time-series. Additionally, we investigate the effect of possible signal redundancy on classification scores and computational performance. This will be investigated using the raw EEG data rather than splitting it into different frequency bands, therefore eliminating data preparation steps.

Results show that with both channel selection and sample reduction methods, we could accomplish comparable classification results with KNN, Support Vector Machines (Classifier: SVC), KNN and RF when run on the entire dataset containing signals from all channels. Additionally, outcomes suggest that significant computational speed-up could be achieved using a Mutual Information (MI) measure for EEG

2 DATASET

The data corpus explored in this work was collected and compiled by Roesler [15], and provided for open access on UCI data repository[12]. The dataset comprises of raw electro-magnetic recordings taken from the scalp of one participant and information about eye-state (eyes open or closed) over the same time period. The participant was asked to relax, look forwards towards a camera and blink naturally, without restriction [9]. While looking toward the camera a video was recorded of the eye. Once recorded the video data was coded; binary labels were used to identify the two different eye-states; '1' for an "eye-blink" and '0' for "eye-open" state — the distribution of eye-states over the course of the recording in the dataset is shown in Fig. 2.

During this time, recordings were also taken at the scalp using the Emotiv EEG Neuroheadset, which measured the electro-magnetic signal at 14 electrode positions (see Fig. 3 - note that two electrode positions where excluded, as indicated in the figure). 14980 sequential timepoints (observations) were recorded from each of the 14 EEG channels (features), The recording took place over 117 seconds period (this is a rate of 128Hz) and measured signals were stored as floating-point values.

Initial exploration of the dataset indicated three outliers (value > 10x the average recording) , which were removed. Observations were therefore reduced to 14977 at each electrode; each of these 14 timeseries represented the signal variability of an electrode over the experimental period. The timeseries were then normalised using zero centring ("de-meaning" applied) to explore the positive and negative deviation from their mean, as an indicator of signal similarity — centred signals are shown in Fig.4 . By eye, the timeseries show overall similarity across the different electrodes.

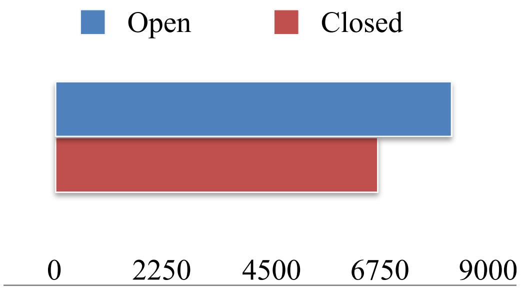


Figure 2: Eye-state distribution of the recorded observations.

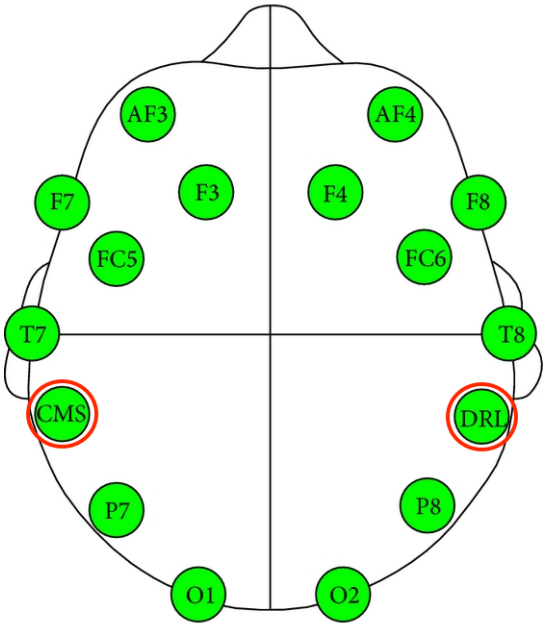
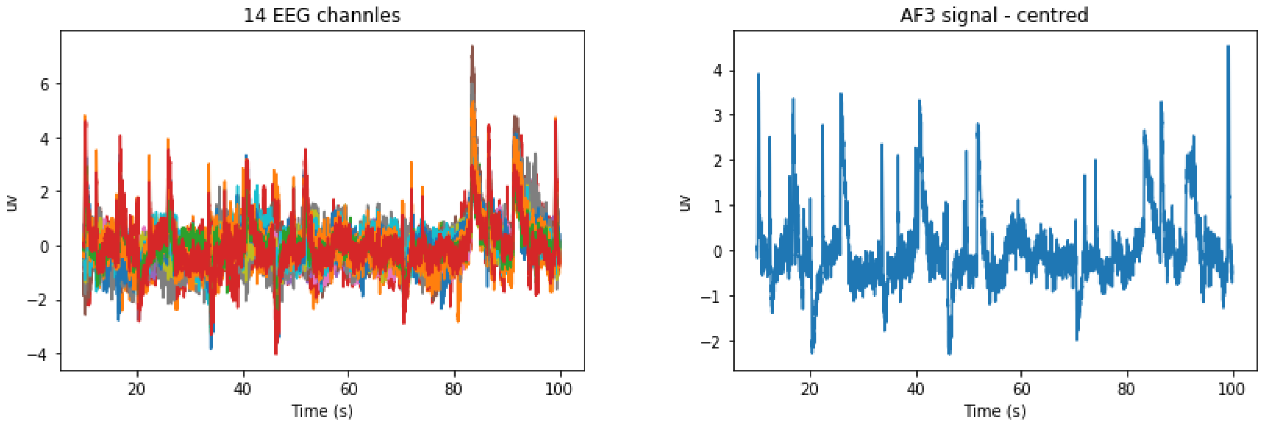


Figure 3: The 14 EEG channels compiled by Roesler[15]. Excluded channels are circled in red.

For example, a strong signal similarity can be observed when looking at the time-series of AF3 and F7 as in Fig. 5.

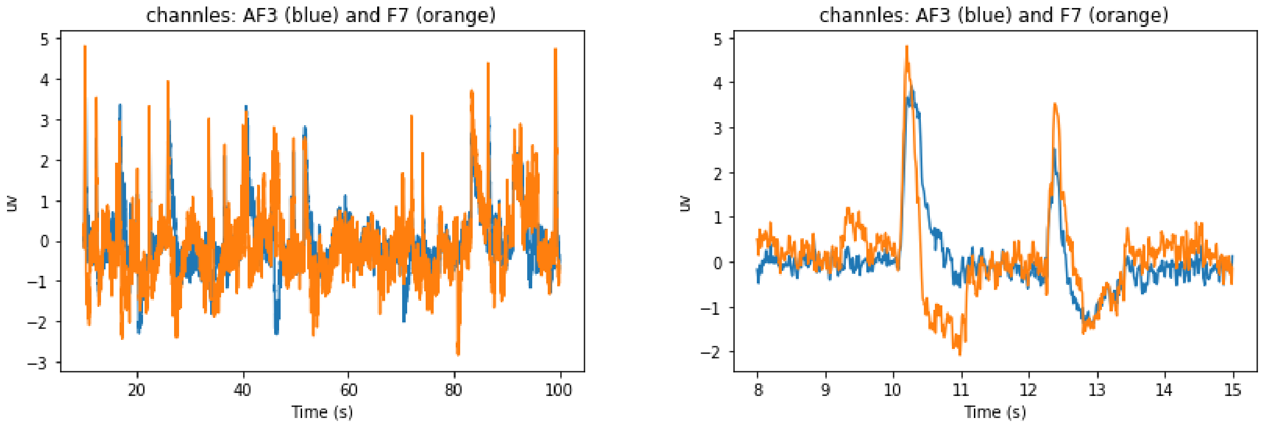
Similarities across the EEG timeseries are observable overall. Indeed, if we separate all EEG time-series relative to eye-state and cross-correlate, topological patterns in signal variability across the channels exist, see Fig. 6 (note that order of variables varies across the matrices to facilitate visualisation of possible similarities). The correlation between electrode signals is also seen to change during eyes-close state compared to eyes-open.



(a) 14 channel time-series

(b) 1 channel time-series

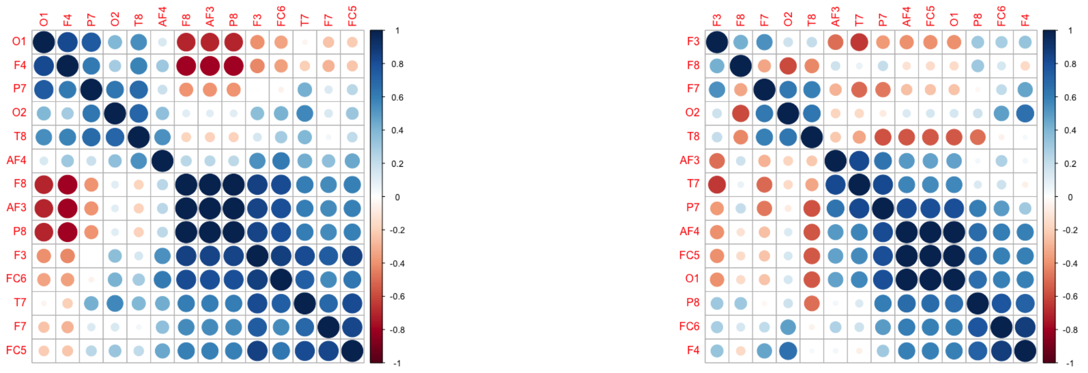
Figure 4: (a) Sliced window of the 14 EEG channel centered time-series- um is signal voltage. Signals show similarity across the EEG channels. (b) Only one channel (AF3) signal is shown.



(a)

(b)

Figure 5: Signal time-series for AF3 and F7, (b) shows a smaller time-slot of the same time-series.



(a) eye-state=1

(b) eye-state=0

Figure 6: Correlation matrices for an 'eye-blink' time-series in (a), and 'eye-open' in (b). Hierarchical clustering [11] is applied to cluster higher-correlated channels together. Note the difference of order in each matrix.

3 METHODS AND FINDINGS

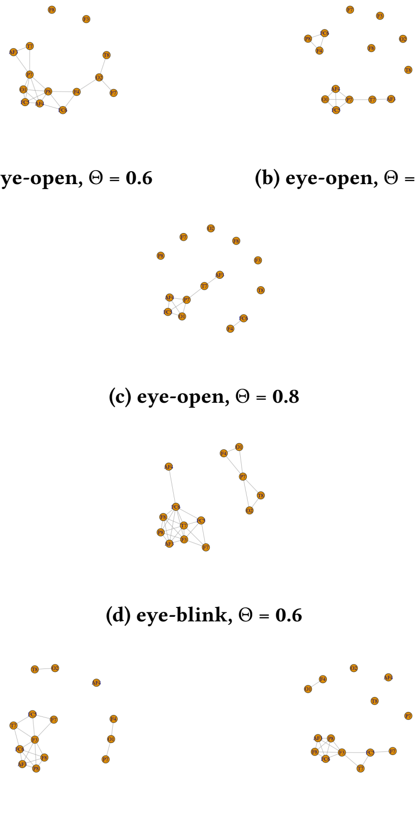
Patterns between EEG channel signals relative to eye-state could be further explored by techniques from graph theory. Linear correlations between the time series $T_i(t_k)$ and $T_j(t_k)$ (the Pearson correlation coefficient R_{ij}) given by

$$R_{ij} = \frac{\sum\limits_{k=1}^{L} T_i(t_k)T_j(t_k)}{\sqrt{(\sum\limits_{k=1}^{L} T_i^2(t_k))(\sum\limits_{k=1}^{L} T_j^2(t_k))}} \tag{1}$$

is widely used [20], whereby strong linearity between two channels can be expressed as a link between two graph nodes. Having derived the correlation matrix C, a threshold τ is usually applied to define strong similarities between graph nodes as 'links'. The adjacency matrix A for the graph is then found by

$$A_{ij} = A_{ji} = \Theta(C_{ij} - \tau) - \delta_{ij}, \tag{2}$$

where Θ is the Heaviside function and δ is Kronecker delta. Graphs based on the two different eye-states have been constructed, considering different values for Θ — see Fig. 7.



(a) eye-open, Θ = 0.6

(b) eye-open, Θ = 0.7

(c) eye-open, Θ = 0.8

(d) eye-blink, Θ = 0.6

(e) eye-blink, Θ = 0.7

(f) eye-blink, Θ = 0.8

Figure 7: Different graphs constructed from 14 EEG channel time-series, relative to eye-state and a value for Θ; strength of linear similarity.

The constructed graphs (small, provided the number of channels) show a strong dissimilarity in topological structure between first set (eye-open signals) and the second (constructed from eye-blink signals). Studying metrics such as the average degree for nodes of both types of graphs can be used to quantify the topological similarity further.

Based on observed similarities (linear similarity explored here), we argue that machine learning methods should provide comparable results if, on the one hand features' space is reduced based on the relevance of features and their 'score' of redundancy, and on another, the similarity between signals can be captured in shorter time-series of signals. To test both assumptions we sliced the provided time-series for all channels into time-windows of 3 seconds (384 timepoints, collected at a rate of 128Hz), containing a transition between eye-blink and eye-open and vice versa. — Fig 8 shows a time-series window of 7 seconds for demonstration purposes.

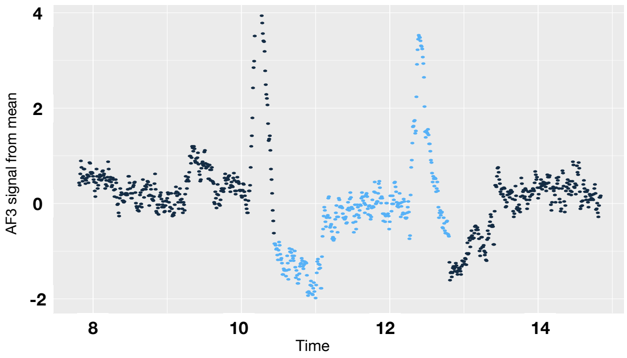


Figure 8: Example of F7 signal time-series window — window here is of 7 seconds. Light blue represent recordings in the closed-eye state and dark blue eyes-open.

20 time-series slices (windows of 3s length each) were generated for each channel, resulting in the total number of observations reduced to 7,680. We then implemented a filtering approach based on mutual information (M_{ij}), given by

$$M_{ij} = \sum_{T_i,T_j} P_{ij}(T_i,T_j) \log \frac{P_{ij}(T_i,T_j)}{P_i(T_i)P_j(T_j)}. \tag{3}$$

Here $P_i(T_i)$ is the probability density function (PDF) of time series T_i, and $P_{ij}(T_i,T_j)$ is the joint PDF for (T_i,T_j).

The minimum redundancy maximum relevance (mRMRe) algorithm [4], as a filtering method, uses differences of M_{ij} to compute the degree of dependency between multiple random variables. The method then sequentially compares the relevancy/redundancy balance of information between variables, providing scores for both their relevance and redundancy. The variable (channel time-series) selected at each step is the one with the highest score. A negative score indicates a redundancy final trade of information and a positive score indicates a relevancy final trade of information. The scoring results averaged for the 20 time-series slices is shown in Fig 9.

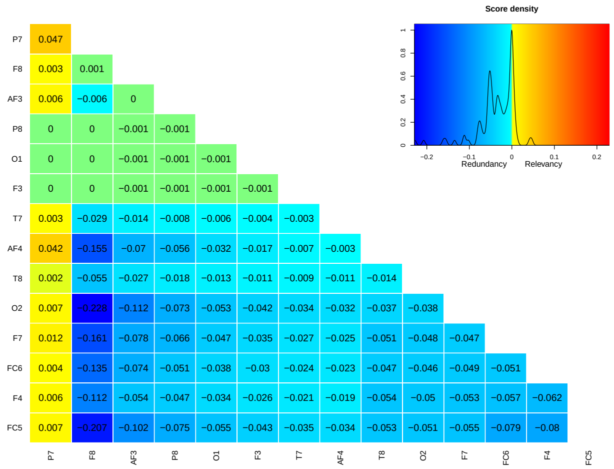


Figure 9: The variable selected at each step is the one with the highest scores (the variables are ordered in the plot). The scores at selection can thus be read from the diagonal.

Accordingly we ran a series of experiments on a 3.2 GHz Intel Core i5 processor with 16 GB 1600 MHz DDR3 memory, which included a base run of different machine learning methods for classification; Nearest neighbors (KNN), Logistic Regression, Support Vector Machines classifier (SVC) and Random Forests (RF). The base run included the tuning of K for nearest neighbors and regularisation, using grid search methods, for SVC hyper-parameters (C=10, gamma=0.001) with radial basis function (rbf) kernel. RF is run with bootstrapping enabled and the number of selected features set to 'automatic', in order to decrease variance amongst constructed member trees.

K-fold validation score of F1 metric was obtained over the entire dataset of 14977 time-series (117 seconds and sampling rate of 128Hz). Although F1 score's value is not the main concern here, a comparison between classification methods' accuracy can be found in [15] and in [14] for deep learning architectures' performance. The resulting classification F1 Score and processing speed where then compared to those of 3 different experimental settings, for KNN, Logistic Regression, SVC and RF, which included:

- (A) 9 features from mRMRe analysis are selected from the available 14. Entire dataset of 14977 time-series is used to learn the classifiers. Accuracy and performance results compared to base run are shown in Table 1 below:

Classifier	F1 Score gain	Speed-up gain
KNN	-0.2	2.1x
LogReg	-0.36	2x
SVC	-0.3	3x
RF	-0.5	0.3x

Table 1: Accuracy and processing time comparison between base run and experiment A.

- (B) 9 features from mRMRe analysis are selected from the available 14. 7,680 observations (based reducing time-series to 20 sliced windows containing eye-state transition) used for learning. Accuracy and performance results compared to base run are shown in Table 2 below:

Classifier	F1 Score gain	Speed-up gain
KNN	-0.4	4.3x
LogReg	-0.6	2.5x
SVC	-0.7	5.6x
RF	-0.7	3x

Table 2: Accuracy and processing time comparison between base run and experiment B.

- (C) 14 features (no channel selection) on the 7,680 observations are used to learn a classifier. Accuracy and performance results compared to base run are shown in Table 3 below:

Classifier	F1 Score gain	Speed-up gain
KNN	-0.1	not observed
LogReg	-0.17	2.3x
SVC	-0.3	1.9x
RF	-0.63	3x

Table 3: Accuracy and processing time comparison between base run and experiment C.

The obtained results show some classification score penalty in most runs of the presented methods, generally, yet speed-up gain is promising. The accuracy score declines more as both feature reduction as well as data slicing method are applied together, however, processing speed-up gains are maximised. Further tuning to, namely, SVC is believed to show better scores by the application of this work's methods, which we aim at focusing on for future developments. That said, the method introduced can be vastly useful for the analysis of higher-dimensional EEG/MEG signals which are typically characterised by the existence of both redundancy of information in their signals, and most importantly, noise.

4 CONCLUSIONS AND FURTHER WORK

This paper presented a brief literature on the analysis of the electroencephalogram (EEG) signals and the application of their analysis. Information obtained by Emotiv headsets on subjects/humans include signal time-series from different electrodes, which typically exhibit variability, in response to events designed for a study of interest. The resolution of collected data as well as the quantity of time-series which could be obtained by such devices is, increasingly, producing both opportunities to gain further insights into brain functionality, and a challenge on the analysis side; accuracy and efficiency. In the presented work, we showed that efficiency could be improved with some (arguably marginal) penalty on a range of popular machine learning accuracy outcomes that are applied for the analysis of EEG data. The introduced method assumes that much of EEG signal information can be captured by (A) signals in a subset of EEG channels, which we filtered by the application of mRMRe technique and (B) signal information from discrete time-series slices (3secs) which contain signal (eye-state) transitions. Experimental results obtained show that both assumptions hold for the classification of eye-state from 14 EEG channels based on the dataset provided by Roesler, see Section 2.

When developing this work, results should be considered alongside deep neural network architectures which have shown low convergence times and may be applicable in real-time classification of eye-state [14].

Slicing frequency and the number of features/channels to select have been done heuristically here, and therefore, based on these preliminary outcomes, we hope to validate the presented method and obtained outcomes on larger datasets, in future work. Also, we believe that more noticeable gains in learning efficiency should be possible for datasets of significantly higher-dimensional features' spaces.

CODE AVAILABILITY

Authors will provide link to code and dataset in camera-ready manuscript.

REFERENCES

[1] Turky Alotaiby, Fathi E. Abd El-Samie, Saleh A. Alshebeili, and Ishtiaq Ahmad. 2015. A review of channel selection algorithms for EEG signal processing. *EURASIP Journal on Advances in Signal Processing* 2015, 1 (01 Aug 2015), 66. https://doi.org/10.1186/s13634-015-0251-9

[2] Muhammad Zeeshan Baig, Nauman Aslam, Hubert P.H. Shum, and Li Zhang. 2017. Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery EEG. *Expert Systems with Applications* 90 (2017), 184 – 195. https://doi.org/10.1016/j.eswa.2017.07.033

[3] Robert J Barry, Adam R Clarke, Stuart J Johnstone, Christopher A Magee, and Jacqueline A Rushby. 2007. EEG differences between eyes-closed and eyes-open resting conditions. *Clin Neurophysiol* 118, 12 (Dec 2007), 2765–2773. https://doi.org/10.1016/j.clinph.2007.07.028

[4] R. Battiti. 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks* 5, 4 (1994), 537–550. https://doi.org/10.1109/72.298224

[5] Hans Berger. 1931. Über das Elektrenkephalogramm des Menschen. *Archiv für Psychiatrie und Nervenkrankheiten* 94, 1 (01 Dec 1931), 16–60. https://doi.org/10.1007/BF01835097

[6] Maria Boersma, Dirk J A Smit, Henrica M A de Bie, G Caroline M Van Baal, Dorret I Boomsma, Eco J C de Geus, Henriette A Delemarre-van de Waal, and Cornelis J Stam. 2011. Network analysis of resting state EEG in the developing young brain: structure comes with maturation. *Hum Brain Mapp* 32, 3 (Mar 2011), 413–425. https://doi.org/10.1002/hbm.21030

[7] Andrea Brovelli, Piero Paolo Battaglini, Jose Raul Naranjo, and Riccardo Budai. 2002. Medium-Range Oscillatory Network and the 20-Hz Sensorimotor Induced Potential. *NeuroImage* 16, 1 (2002), 130 – 141. https://doi.org/10.1006/nimg.2002.1058

[8] Michele Ferrara and Luigi De Gennaro. 2011. Going Local: Insights from EEG and Stereo-EEG Studies of the Human Sleep-Wake Cycle. *Current topics in medicinal chemistry* 11 (08 2011), 2423–37. https://doi.org/10.2174/156802611797470268

[9] D. Garrett, D. A. Peterson, C. W. Anderson, and M. H. Thaut. 2003. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11, 2 (2003), 141–144. https://doi.org/10.1109/TNSRE.2003.814441

[10] M Haak, S Bos, Sacha Panic, and Léon Rothkrantz. 2009. Detecting Stress using Eye Blinks during Game Playing. *10th International Conference on Intelligent Games and Simulation, GAME-ON 2009*, 75–82.

[11] Catherine B Hurley. 2004. Clustering Visualizations of Multidimensional Data. *Journal of Computational and Graphical Statistics* 13, 4 (2004), 788–806. https://doi.org/10.1198/106186004X12425 arXiv:https://doi.org/10.1198/106186004X12425

[12] School of Information Irvine, CA. University of California and Computer Science. [n.d.]. UCI Data Repository. https://archive.ics.uci.edu/ml/datasets/

[13] W J Levy. 1987. Effect of epoch length on power spectrum analysis of the EEG. *Anesthesiology* 66, 4 (Apr 1987), 489–495.

[14] T. K. Reddy and L. Behera. 2016. Online Eye state recognition from EEG data using Deep architectures, In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 000712–000717. https://doi.org/10.1109/SMC.2016.7844325

[15] Oliver Rösler and David Suendermann. 2013. A First Step towards Eye State Prediction Using EEG. In *International Conference on Applied Informatics for Health and Life Sciences (AIHLS 13)*.

[16] R Sandyk. 1990. The significance of eye blink rate in parkinsonism: a hypothesis. *Int J Neurosci* 51, 1-2 (Mar 1990), 99–103.

[17] C.J. Stam, T. Montez, B.F. Jones, S.A.R.B. Rombouts, Y. van der Made, Y.A.L. Pijnenburg, and Ph. Scheltens. 2005. Disturbed fluctuations of resting state EEG synchronization in Alzheimer's disease. *Clinical Neurophysiology* 116, 3 (2005), 708–715. https://doi.org/10.1016/j.clinph.2004.09.022

[18] Bhawna Vermani, Neha Hooda, and Neelesh Kumar. 2015. *Parametric evaluation of EEG signal during Eyes Close and Eyes Open state.* 1–5 pages. https://doi.org/10.1109/INDICON.2015.7443754

[19] T. Wang, S. U. Guan, K. L. Man, and T. O. Ting. 2014. Time Series Classification for EEG Eye State Identification Based on Incremental Attribute Learning, In 2014 International Symposium on Computer, Consumer and Control. *2014 International Symposium on Computer, Consumer and Control*, 158–161. https://doi.org/10.1109/IS3C.2014.52

[20] Daniel S. Wilks. 2011. *Statistical Methods in the Atmospheric Sciences.* Elsevier. http://books.google.com/books?hl=en&lr=&id=IJuCVtQ0ySIC&oi=fnd&pg=PP2&dq=Statistical+Methods+in+the+Atmospheric+Sciences&ots=anFosWzIJ_&sig=kCqI0fmU8UjjV5ZtVlifGWGcSJc

[21] Jiahui Xu and Baichang Zhong. 2018. Review on portable EEG technology in educational research. *Computers in Human Behavior* 81 (2018), 340 – 349. https://doi.org/10.1016/j.chb.2017.12.037

[22] Jianhua Yang, Harsimrat Singh, Evor L. Hines, Friederike Schlaghecken, Daciana D. Iliescu, Mark S. Leeson, and Nigel G. Stocks. 2012. Channel selection and classification of electroencephalogram signals: An artificial neural network and genetic algorithm-based approach. *Artificial Intelligence in Medicine* 55, 2 (2012), 117 – 126. https://doi.org/10.1016/j.artmed.2012.02.001

[23] Mervyn V.M. Yeo, Xiaoping Li, Kaiquan Shen, and Einar P.V. Wilder-Smith. 2009. Can SVM be used for automatic EEG detection of drowsiness during car driving? *Safety Science* 47, 1 (2009), 115 – 124. https://doi.org/10.1016/j.ssci.2008.01.007

Ant-Driven Clustering for Utility-Aware Disassociation of Set-Valued Datasets

Nancy Awad
Femto-ST Institute, UMR 6174 CNRS,
University of Bourgogne-Franche-Comte
Belfort, France
TICKET Lab., Antonine University
Hadat-Baabda, Lebanon
nancy.awad@ua.edu.lb

Jean-François Couchot
Femto-ST Institute, UMR 6174 CNRS,
University of Bourgogne-Franche-Comte
Belfort, France
jean-francois.couchot@univ-fcomte.fr

Bechara Al Bouna
TICKET Lab., Antonine University
Hadat-Baabda, Lebanon
bechara.albouna@ua.edu.lb

Laurent Philippe
Femto-ST Institute, UMR 6174 CNRS,
University of Bourgogne-Franche-Comte
Besançcon, France
laurent.philippe@univ-fcomte.fr

ABSTRACT

Data publishing is a challenging task from the privacy point of view. Different anonymization techniques are proposed in the literature to preserve privacy in accordance with some mathematical constraints. Disassociation is one of the anonymization techniques that relies on the $k^m - anonymity$ privacy constraint to guarantee a certain level of privacy for set-valued datasets (*e.g.*, search and shopping items). Disassociation separates a set-valued dataset by clustering the dataset into groups of records with common frequent items, and then splitting each cluster into record chunks respecting $k^m - anonymity$. In this paper, we define a new ant-based clustering algorithm based on the disassociation technique to keep some of the items associated together throughout the anonymization process. We define these associations as utility rules that should be treated with eagerness while anonymizing the data. We perform a set of experiments to evaluate our algorithm w.r.t. these utility rules.

CCS CONCEPTS

• **Security and privacy** **Usability in security and privacy**; *Data anonymization and sanitization*; Privacy-preserving protocols; • **Computing methodologies** *Artificial life*.

KEYWORDS

anonymization, utility, privacy, disassociation, ant colony clustering

ACM Reference Format:
Nancy Awad, Jean-François Couchot, Bechara Al Bouna, and Laurent Philippe. 2019. Ant-Driven Clustering for Utility-Aware Disassociation of Set-Valued Datasets. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3331076.3331084

1 INTRODUCTION

Data publishing has become a challenging task considering all the disciplines that are involved in the process. In fact, the privacy of the data is a major concern that increases with the complexity and the size of the data. Many are the anonymization techniques, presented in the literature, that can be employed to protect the privacy of the users in the data [4, 8, 15, 19, 21, 24, 25, 27]. However, anonymizing the data is a burden on the utility. Set-valued data provides enormous opportunities for various data analysis, for that reason, a trade-off between data privacy and data utility must be found. The aim is not only to provide a good anonymization of the data but also to make the output valuable for future analysis. In this work, we focus on one technique: disassociation, presented in [22], which divides a set-valued dataset into clusters and then each cluster into record chunks preserving the $k^m - anonymity$ privacy constraint. Disassociation is studied in [2] where a privacy breach defined as the cover problem was found. We propose a solution for this cover problem in [1]. In this work, we study the problem of publishing predefined set of associations, we call utility rules, under the disassociation technique and with rigorous attention to their utility. Actually, the horizontal partitioning in disassociation groups data records in clusters using a naive similarity function between the records. Under this perspective, we propose a variation of ant-based clustering methodology, to increase the utility of predefined associations. Swarm intelligence is the domain of studying the social behaviors of swarms like ant colonies, bird and fish schools. For more than three decades, swarm intelligence has been flourishing and used effectively in multiple fields to solve optimization problems like the traveling salesman problem using the ant colony optimization (ACO) [7], constructing portfolios of stock in the financial field using particle swarm optimization (PSO) [28] and finding the best position to hide information using

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06.
https://doi.org/10.1145/3331076.3331084

Table 1: Notations used in the paper

$\mathcal{D}$	a set of items
$\mathcal{T}$	a table containing individuals related records
$\mathcal{T}^*$	a table anonymized using the disassociation technique
r	a record (of $\mathcal{T}$) which is set of items associated with a specific individual of a population
I	an itemset included in $\mathcal{D}$
$s(I, \mathcal{T})$	the number of records in $\mathcal{T}$ that are superset of I
C	a cluster in a disassociated dataset, formed by the horizontal partitioning of $\mathcal{T}$
C^*	a vertically partitioned cluster C that results in record chunks and a term chunk
R_C	a record chunk from the vertically partitioned cluster C^*
T_C	the term chunk from the vertically partitioned cluster C^*
δ	maximum number of records allowed in a cluster, also know as the maximum cluster size

the cat swarm optimization (CSO) [26]. Through the observation of the collective behaviors of decentralized, self-organized natural systems, it is fascinating to discover how with limited individual abilities, swarms working together, can accomplish complex tasks. Inspired by the behavior of real ants and their pheromone-based communication, we present a variation of ant-based algorithm to cluster the data related to the utility rules for the disassociation.

The rest of the paper is organized as follows: First, Section 2 describes the problem of the data utility preservation under a specific anonymization technique the disassociation, and evaluate it theoretically. Then, Section 3 recalls some of the traditional and the swarm intelligence based data clustering methods with their advantages and limitations. To solve our problem, we propose a variation of an ant-based clustering technique described in Section 4. Finally, Section 5 investigates the efficiency of our solution and its impact on the utility of aggregate analysis for a predefined set of association rules. We conclude the article in Section 6 and present outlines for future work.

2 PROBLEM DEFINITION AND BACKGROUND

2.1 Problem definition

The disassociation technique, as proposed in [22], is driven by the idea of ensuring the k^m – *anonymity* privacy constraint by separating the terms of a record in multiple record chunks within a cluster. This process creates ambiguity for an association between its separated terms, which causes a reduction of the utility for the association in question. Disassociation works on the assumption that there are no specific associations more valued than others and that data items must not be altered, generalized or suppressed. In this paper, our aim is to provide a better utility for a set of predefined associations, called the utility rules, by reducing the amount of split-ups a utility rule has to endure in order to preserve k^m – *anonymity*. Formally, k^m – *anonymity* is defined as follows:

Definition 2.1 (k^m – anonymity). Given a dataset of records $\mathcal{T}$ whose items belong to a given set of items $\mathcal{D}$. The dataset $\mathcal{T}$ is k^m – *anonymous* if $\forall I \subseteq \mathcal{D}$ such that $|I| \leq m$, the number of records in $\mathcal{T}$ that are superset of I is greater than or equal to k, *i.e.*, $s(I, \mathcal{T}) \geq k$.

In what follows, we review the disassociation technique under the perspective of the utility of associations. Table 1 recalls the basic notations used in this paper.

2.2 Disassociation and utility awareness

In this work, we are interested in one anonymization technique, the disassociation, which relies on k^m – *anonymity* to guarantee the data privacy. We dedicate this section to show how this technique is a two-sided coin for publishing data:

• the first side of the coin is concerned with the data utility and relies on the record's clustering,

• the second side of the coin is involved in attaining a privacy level through terms' split-ups.

We use Fig-1 to illustrate an example of disassociation, applied with $k = 3$, $m = 2$ and $\delta = 6$.

	$\mathcal{T}$
r_1	a b c d e
r_2	a b c
r_3	a d e
r_4	a d
r_5	a e
r_6	a
r_7	f b c
r_8	f g b
r_9	f g c
r_{10}	f g
r_{11}	g b

(a) Original Dataset $\mathcal{T}$

C_1
a b c d e
a b c
a d e
a d
a e
a

C_2
f b c
f g b
f g c
f g
g b

(b) Horizontal Partitioning

C_1^*		
R_{C_1}	R_{C_2}	TC
a d	e	b
a		c
a d	e	
a d		
a	e	

C_2^*		
R_{C_1}	R_{C_2}	TC
f	b	c
f g	b	
f g		
f g		
g	b	

(c) Vertical Partitioning

Figure 1: Example of disassociation

C_1
a b c d e
a b c
f b c

(a) Horizontal Partitioning for ur

C_1^*	
R_{C_1}	TC
b c	a
b c	d
b c	f
	e

(b) Vertical Partitioning

Figure 2: Example of a utility driven disassociation

Horizontal partitioning as presented in [22], clusters the records using a naive similarity function. Records are grouped together in

clusters of maximum size equal to δ, based on a common frequent term. The authors of [22] justify the use of the naive method to two main complaints regarding the similarity functions: first, they are inefficient on large datasets, and second, they do not explicitly control the size of the clusters. However, this process of horizontal partitioning based on the support of items, doesn't take into account the associations in the data.
Horizontal partitioning as shown in Fig-1(b), groups records that contain the most frequent item, a having $s(a, \mathcal{T}) = 6$, within cluster C_1, and all the other records within C_2. Both clusters have a size less than δ, then C_1 and C_2 are vertically partitioned.

Vertical partitioning is the process of creating for each cluster, record chunks that verify the $k^m - anonymity$ privacy constraint, and a term chunk that contains the items having a support less than k in the cluster.
In our example, vertical partitioning is applied over each cluster returned by the horizontal partitioning, C_1 and C_2, splitting the items into different record chunks with respect to the $k^m - anonymity$ privacy constraint. The final result of disassociation is shown in Fig-1(c).

To illustrate the problem of utility, let's consider that the frequency of the association $\{b, c\}$, is important in the analysis after data publishing. Unfortunately, we can barely extract valuable information about the association between items b and c from Fig-1(c). In C_1^* both items are added to the term chunk because their support was less then $k = 3$, showing neither the association between them nor their real support. Similarly, association $\{b, c\}$ is unclear in C_2^*, with only one advantage over C_1^*; knowing the support of item b.
Let's suppose that there exists a clustering technique that favors the association $\{b, c\}$ while disassociating $\mathcal{T}$, and brings together all the records related to it as in Fig-2(a). After applying vertical partitioning in Fig-2(b), the association $\{b, c\}$ is totally preserved associated. Now, any analysis over the support of $\{b, c\}$ is accurate. Hence, data utility depends essentially on horizontal partitioning.
From this example, we deduce that the need to give a data analyst the ability to define a set of associations, we call utility rules, that are important in future analysis is crucial. Those utility rules must be preserved carefully within the anonymized data.

2.3 Utility rules

Giving an exact general definition for the utility of the data in the domain of anonymization is irrational. Generally, the utility is the quality of data for the intended use and it expends within different definitions in the literature. Some works consider the utility as a practical guide to reduce the extent of data generalization [11] whereas in [9] a clustering based technique is implemented to minimize the abstraction. In this work data utility, is considered for the aggregate query answering accuracy. Frequent or not, we assume that any utility rule should be well represented after disassociation. Let us first formalize our utility rules and their context:
• Let $\mathcal{T} = \{r_1, ..., r_n\}$ be the original dataset of records. Every record $r_i \in \mathcal{T}$ is a set of items $r_i = \{y_1, ..., y_p\}$ related to a specific individual.
• Let $UR = \{ur_1, ..., ur_u\}$ be a dataset of predefined associations, where each utility rule $ur_i = \{x_1, ..., x_d\}$ is a set of items from the same domain of $\mathcal{T}$. In this work we consider that every utility rule exists at least once in the original dataset: $\forall\ ur_i \in UR, \exists r \subseteq \mathcal{T}$ such that $ur_i \subseteq r$.
• Let $s(ur, \mathcal{T})$ be the support of the utility rule ur from UR in the original dataset, which is the number of records from $\mathcal{T}$ containing ur.
• Let $s(ur, \mathcal{T}^*)$ be the support of the utility rule in a disassociated dataset.
• Let $conf(ur, \mathcal{T})$ be the confidence of the utility rule ur in the original dataset $\mathcal{T}$, defined as:

$$conf(ur, \mathcal{T}) = \frac{s(ur, \mathcal{T})}{|\mathcal{T}|}$$

• Let $conf(ur, \mathcal{T}^*)$ be the confidence of the utility rule ur in the disassociated dataset $\mathcal{T}^*$, defined as:

$$conf(ur, \mathcal{T}^*) = \frac{s(ur, \mathcal{T}^*)}{|\mathcal{T}^*|}$$

We know that $\mathcal{T}^*$ can have a maximum size equal to that of the original dataset, $\mathcal{T}$, and for the following analysis we consider:

$$conf(ur, \mathcal{T}^*) = \frac{s(ur, \mathcal{T}^*)}{|\mathcal{T}|}$$

Definition 2.2 (α − confidence). We say that a utility rule ur is $\alpha - confident$ with:

$$\alpha = \frac{conf(ur, \mathcal{T}^*)}{conf(ur, \mathcal{T})}$$

We use the term confidence, to determine the strength of the association between the items of a utility rule ur after disassociation. Statistical queries are based on the support of the associations in question. The $\alpha - confidence$ represents a percentage of the original support of a utility rule that must be reflected in the final output of the disassociation for the utility rule.
In what follows we evaluate theoretically the utility of an association under $k^m - anonymity$.

2.4 The utility privacy trade-off in disassociation

The privacy constraint of disassociation forces any output of the data to be $k^m - anonymous$ in every record chunk and no cluster can be larger than δ. In what follows, we evaluate $\alpha - confidence$ under the $k^m - anonymous$ privacy context of disassociation.
We formalize our context for the following analysis:
• Let R_{i_C} be a record chunk from the vertically partitioned cluster C_i^*.
• Let $X = argmin_{I \subseteq ur, I \in R_{i_C}} s(I, R_{i_C})$ be the subset of the utility rule ur, found in a record chunk R_{i_C} from cluster C_i^*, having the minimum support between all subsets of the different record chunks.
• Let $Y = \{y | y \subset ur, y \in R_{i_C}\} \setminus X$ be the set of the subsets of the utility rule ur, except X, found in the record chunks of the cluster C_i^*.
• Let $Z = \{e | e \in TC, e \in ur\}$ be the set of items of the utility rule ur that belong to the term chunk TC of the cluster C_i^*.

Definition 2.3. We assume that the support of a utility rule, $ur = X \cup Y \cup Z$, when vertically partitioned in C_i^*, is the average support of reconstructing it, $avgs(ur, C_i^*)$, calculated as the product of the minimum possible support by the frequencies of all the other subsets of *ur* in the record chunks:

$$avgs(ur, C_i^*) = \gamma * \prod_{j=1}^{p}(fr(y, R_{i_{C_j}}))$$

where:

• $fr(y, R_{i_{C_j}})$ is the frequency of any subset of ur, $y \in Y$, present in a record chunk of C_j^* defined as:

$$fr(y, R_{i_{C_j}}) = \frac{s(y, R_{i_{C_j}})}{|R_{i_{C_j}}|}$$

• γ represents the maximum possible original support of *ur* in C_i. In fact, a utility rule cannot appear more than any of its subsets in the record chunks, thus γ is the minimum subset of *ur* in C_i^* :

$$\gamma = \begin{cases} s(X, C_i^*) \text{ if } Z = \emptyset \\ \min(2^{|TC|-|Z|}, k-1) \text{ if } Z \neq \emptyset \end{cases}$$

with $2^{|TC|-|Z|}$ representing the number of possibilities for reconstructing Z in the term chunk TC

• p is the number of record chunks that contain subsets of *ur*.

Definition 2.4. The support of *ur* in the whole disassociated dataset,$\mathcal{T}^*$, is the sum of the average support of *ur* in every vertically partitioned cluster:

$$s(ur, \mathcal{T}^*) = \sum_{i=0} avgs(ur, C_i^*)$$

LEMMA. *$\forall ur \in UR$, a $k^m -$ anonymous disassociation ensures $\alpha -$ confidence for ur, where:*

$$\frac{1}{\delta^{|ur|}} \leq \alpha \leq 1$$

PROOF 1.

$k^m -$ anonymity ensures that any subset in a record chunk is present at least k times when its cardinality is less than or equal to m. If its cardinality is greater than m, it may be present one time in a record chunk and verify the $k^m -$ anonymity constraint. Furthermore, a record chunk contains at most δ records due to the maximum cluster size constraint. Then:

$$k \leq s(X, R_{i_C}) \leq \delta,\ if\ |X| \leq m$$
$$1 \leq s(X, R_{i_C}) \leq \delta,\ if\ |X| > m$$

With the same reasoning, we know that the frequency of y in R_C, is bounded by:

$$\frac{k}{\delta} \leq fr(y, R_{i_{C_j}}) \leq \frac{\delta}{\delta},\ if\ |y| \leq m$$
$$\frac{1}{\delta} \leq fr(y, R_{i_{C_j}}) \leq \frac{\delta}{\delta},\ if\ |y| > m$$

A utility rule can be retrieved in at most $|ur|$ record chunks, when every item from ur is present in a distinct record chunk after vertical partitioning, then $\forall y \in Y$:

$$(\frac{1}{\delta})^{|ur|-1} \leq \prod_{j=1}^{|ur|} fr(y, R_{i_{C_j}}) \leq 1$$

When Z is reconstructed from the term chunk, TC, it cannot have a size greater to $k - 1$, hence:

$$1 \leq min(2^{|TC|-|Z|}, k-1) \leq k-1$$

To generalize, we consider that the least frequent subset of ur is present in a record chunk and not in the term chunk with $Z = \emptyset$, then γ of definition 2.3 is bounded by:

$$1 \leq \gamma \leq \delta$$

We can deduce from definition 2.3 that:

$$(\frac{1}{\delta})^{|ur|-1} \leq avgs(ur, C_i^*) \leq \delta \qquad (1)$$

From definition 2.4, we calculate the support of ur in $\mathcal{T}^*$ by adding the average support through the clusters that represent ur. In fact, we can reconstruct ur in at least $\frac{s(ur,\mathcal{T})}{\delta}$ clusters. Therefore, we multiply the inequality (1) *by this minimum number of clusters representing ur:*

$$\frac{s(ur, \mathcal{T})}{\delta^{|ur|}} \leq s(ur, \mathcal{T}^*) \leq s(ur, \mathcal{T})$$

Following definition 2.2, we calculate the $\alpha -$ confidence of ur as:

$$\alpha = \frac{conf(ur, \mathcal{T}^*)}{conf(ur, \mathcal{T})} = \frac{s(ur, \mathcal{T}^*)}{s(ur, \mathcal{T})} * \frac{|\mathcal{T}|}{|\mathcal{T}|} = \frac{s(ur, \mathcal{T}^*)}{s(ur, \mathcal{T})}$$

therefore:

$$\frac{1}{\delta^{|ur|}} \leq \alpha \leq 1$$

From this analysis, we can see that to ensure $k^m -$ anonymity, the confidence of the utility rule can have a very low value. In this case, disassociation provides privacy by putting the utility of data analysis at risk.

The m-α relationship:

It is easy to assume from the above analysis that $\alpha - condidence$ is independent from the m control of the $k^m - anonymity$ constraint. However, this is not true, due to the direct link between m and vertical disassociation. In fact, for $k^m - anonymity$ to be achieved; every association of up to m items should be present at least k times in a record chunk, then if an association cannot be present k times, it must be partitioned over multiple record chunks. Considering all the m possibilities formed from the items of the utility rule ur that will be tested: $\binom{|ur|}{m}$, we can understand how complex it is for a utility rule by its own to withstand the $k^m - anonymity$ test and be preserved non partitioned in one record chunk. Practically, this number of test, $\binom{|ur|}{m}$, is bigger due to other items belonging to the record chunk but not to the utility rule. Then, to be more persistent to the $k^m - anonymity$, a utility rule should not have a very high cardinality, which increases the number of tests to pass with the increase of the number of items forming a utility rule.

We could reflect the m constraint in Proposition-2.4. Eventually $|ur| \leq \max(|ur|, m)$, then:

$$\frac{1}{\delta^{\max(|ur|,m)}} \leq \alpha \leq 1$$

Reaching a more precise lower boundary for $\alpha - confidence$ in the above proof, we hold to that result in Proposition-2.4.

From this perspective of privacy-utility trade-off, we are motivated to contribute with a more insightful horizontal partitioning process, tolerant to the predefined utility rules. Our aim is to disassociate the data, while taking into consideration the utility rules, for future data analysis accuracy. As noticed from the example in Fig-2(b), vertical partitioning is a result of the quality of clustering and is the main privacy pillar of the data. In this paper, we apply the same process of vertical partitioning proposed in [22] and limit our work on the horizontal partitioning. Next section provides a general description of the clustering problem and then the role of swarm intelligence algorithms for the improvement of data clustering.

3 DATA CLUSTERING

Classical clustering:
Clustering is by definition the task of grouping a set of data with similar characteristics together, where data within a cluster are more similar to each other than those in the other clusters. There exists no unified solution to solve all the clustering problems and it has been proven to be NP-hard [23]. We distinguish in the literature two different modes for clustering: fuzzy and partitional. In fuzzy clustering data items may belong to multiple clusters with a fuzzy membership grade like $c - means$ [3]. The previous property does not stand for partitional clustering where clusters are totally disjoint, as in the widely used $K - means$ algorithm [16]. In this work and in accordance with the disassociation principle, we are only interested in partitional clustering.

Swarm Intelligence Clustering:
Originally, Swarm Intelligence (SI) algorithms were adapted in stochastic search and optimization problems. They do not focus on the strict modeling of the natural processes; but use the best ideas to improve the convergence and accuracy of the solutions. Example of SI systems are: ant colony system (ACS) [7, 20], particle swarm optimization (PSO) [13] and artificial bee colony (ABC) [12]. These algorithms draw inspiration from the collective behavior of decentralized, self-organized natural social animals. Even though particles of a swarm may have a very limited individual capabilities, they can perform very complex jobs, vital for their survival, when acting as a community. Diversified jobs like searching and storing food, cleaning corpse and building nests, are wide examples of the complexity of the jobs, performed by the colonies in a perfectionist manner without any kind of supervision. Choosing the right algorithm to solve a problem relies on the comparability of the given problem's background and the swarm's features.

Ant-based Clustering Algorithm (ACA):
Ant-like agents have been applied to solve problems in the context of objects clustering. The inspired ant clustering algorithm (ACA) is modeled after the social behavior of ants sorting larval and cleaning corpse. It was first modeled by [5] to solve robotics' tasks. Two major features influence the action of the ants for picking and dropping items: the similarity and the density of the data within the local neighborhood. From this first model, researches introduced many variation of the algorithm applicable in wider clustering domain to solve different problems [6, 10, 14, 17, 18].

In the next section, we define a variant of the ant-based clustering algorithm to redefine the horizontal partitioning of the disassociation for the predefined utility rules, achieving a higher utility.

Table 2: Ant Colony Terminology

Ant colony	Set of utility rules UR.
Ant	Expert agent a_i working for the benefit of the utility rule ur_i.
Pheromone trail	Square matrix, A, representing the density of each utility rule in each cluster, updated through probabilistic picking-up and dropping functions. It is the sharable memory between the ants.
Food	Data records $\mathcal{T}_{\lceil UR}$ from the dataset $\mathcal{T}$, relative to the utility rules such that: $\mathcal{T}_{\lceil UR} = \{r \in \mathcal{T} \mid \exists ur \in UR \text{ and } ur \subseteq r\}$
Ant's load	$load(a_i)$ contains a data record from $\mathcal{T}_{\lceil UR}$ that the ant a_i is transporting.
Individual ant's job	Picking-up and Dropping a load.

4 UTILITY DRIVEN ANT-BASED CLUSTERING (*UDAC*)

4.1 Framework of the algorithm

In this work, our motivation is to show how a clustering optimization solution can increase the utility value of the predefined set of utility rules in a disassociated dataset. We transform horizontal partitioning for the records related to the utility rules into a clustering optimization problem. The proposed algorithm takes advantage from the widely explored natural ant behaviors. Table 2 describes the environment of our clustering problem in the ant colony system terminology. Our problem is challenging due to the fact that:

- A record might enclose multiple utility rules and since we are working in partitional clustering, this record should belong to exactly one cluster satisfying one utility rule.
- The intersection of terms between the records, can affect the distance metrics.
- The maximum cluster size constant, δ, limits the number of records allowed in a cluster.

Let $\mathcal{T}_{\lceil UR}$ be the set of records from $\mathcal{T}$ concerned with the utility rules UR such that:

$$\mathcal{T}_{\lceil UR} = \{r \in \mathcal{T} \mid \exists\ ur \in UR \textbf{ and } ur \subseteq r\}$$

Let u be the number of utility rules in question:

$$u = |UR|$$

Clusters' initialization:
In this context, only the set of records related to the utility rules, denoted by $\mathcal{T}_{\lceil UR}$, are treated through special clustering. The rest of the records from the dataset $\mathcal{T}$, which aren't supersets of any utility rule, $\mathcal{T} \setminus \mathcal{T}_{\lceil UR}$, are clustered via normal horizontal partitioning. We consider that every utility rule shall be represented in its own cluster and have one ant as its agent. We initialize our algorithm with u clusters and u ants which are the expert agents that will transport the loads from and into the clusters. The algorithm starts by sending its expert ants in search for records containing their representative utility rules from $\mathcal{T}_{\lceil UR}$; recursively until $\mathcal{T}_{\lceil UR}$ is empty, thus distributing the $|\mathcal{T}_{\lceil UR}|$ records along the u clusters. Square matrix $A = [u][u]$, is defined, representing the pheromone trait which is the collective

adaptive memory of the expert ants and is initialized by the value of the support of each utility rule ur_i in each cluster C_j, such that:

$$A[i][j] = s(ur_i, C_j)$$

We denote by β_{ur_i} the ratio of the records representing ur_i in cluster C_i:

$$\beta_{ur_i} = \frac{A[i][i]}{s(ur_i, \mathcal{T})}$$

Expert Ants' Job:
During the clustering process, every ant a_i works for its utility rule ur_i to reach a predefined ratio $\beta_{predefined} \in [0, 1]$, from the original support of the utility rule ur_i, within cluster C_i:

$$\beta_{ur_i} \geq \beta_{predefined}$$

Originally, an ant working in a clustering problem, as defined by [5], picks-up and drops a data object following two probabilistic formulas. The formulas help the ant decide if a data object is dissimilar to its neighborhood and if so, to which cluster it must be transported. In accordance with the literature, our expert ants move loads between the clusters according to the two basic principles, picking-up and dropping loads, adapted to our context:

• *For Pick-Up job:* the expert ant is responsible of choosing a cluster C_j and a record r from it, $r \in C_j$, to transport it into another cluster C_i representing the utility rule ur_i. This pick-up job is controlled by the density of ur_i in the clusters, defined as follows:

$$d(ur_i, C_j) = \frac{A[i][j]}{|C_j|}$$

The expert ant chooses the cluster, C_j, with the highest density of utility rule ur_i:

$$C_j(ur_i) = \underset{\forall j \in [1,\ldots,u]-i}{\arg\max} \quad (d(ur_i, C_j))$$

Then, it transports a record r from the C_j to C_i such that the record r embeds ur_i, $ur_i \in r$.

$$load(a, ur_i) = r \textit{ such that } r \in C_j(ur_i) \textit{ and } ur_i \subseteq r$$

• *For Drop job:* the expert ant a_i is searching for another expert ant, a_j, that needs the most help to increase its β_{ur_j}. We consider that ant a_j is in need for the most help when reaching β_{ur_j} for its utility rule ur_j demands the highest number of iterations:

$$ur_j = \underset{\forall j \in [1,\ldots,u]-i}{\arg\max} \quad (s(ur_j, \mathcal{T}) * (\beta_{predefined} - \beta_{ur_j}))$$

Then, expert ant a_i picks-up a record r for ur_j to add it to its representative cluster C_j following the *Pick-Up* job scenario described above, helping the ant a_j.

Individual and cooperative work of expert ants:
One of the most remarkable trait of any swarm intelligent system, is that the cooperation between the particles of the system leads to the optimized solution despite the limited capabilities of each particle by its own. As discussed before, we recognize this trait in the ant colony clearly when searching for the shortest path leading to food, collecting and grouping corpses.

• If $\beta_{ur_i} \geq \beta_{predefined}$:
To speed up the convergence of the solution and prevent the ants from moving aimlessly, expert ant a_i is now free to help another ant. In this case and in the current iteration, ant a_i stops searching in the space for a record satisfying its utility rule, ur_i. Ant a_i searches for the most vulnerable utility rule according to the *Drop* job scenario, and picks a corresponding record to transport it to the vulnerable cluster, following the *Pick-Up* job scenario.

• If $\beta_{ur_i} < \beta_{predefined}$:
In this case ant a_i is responsible and fully dedicated to its specific utility rule, ur_i. Ant a_i transports a relative record to C_i, according to the *Pick-Up* job scenario, to increase β_{ur_i}. At each iteration β_{ur} is rechecked, for every utility rule ur. In fact, even if at a certain iteration $\beta_{ur} \geq \beta_{predefined}$ this inequality might not stand at the next iteration, due to the exchange of loads that happened during the last iteration.

This cooperation between ants is possible thanks to the collective memory stored in the pheromone trail matrix A. The clustering process is iterative until attaining a predefined number of iterations or a level of stability. In what follows we present the algorithm behind the ant based clustering methodology and explain it thoroughly.

4.2 Our algorithm

In this section, we present our algorithm the Utility-Driven Ant-based Clustering (*UDAC*), that applies the process described above. The algorithm starts by defining the set of records $\mathcal{T}_{|UR}$, from the original dataset $\mathcal{T}$, representing the utility rules (line 1). It continues by creating u clusters, u expert ants and a square matrix $A = [u][u]$ (line 3), all these will be dedicated to represent the u utility rules through the clustering process. Every cluster C_i will be the official nest of exactly one utility rule, ur_i, and expert ant a_i will be working for its benefit while necessary.

At the clusters' initialization phase, successively every expert ant representative of a specific utility rule picks a record from $\mathcal{T}_{|UR}$ that embeds its corresponding utility rule; this process is recursive until emptying $\mathcal{T}_{|UR}$ (lines 5–13). This doesn't mean that the distribution of the records has been fair for the utility rules. In fact, a record may contain multiple utility rules and having one ant transporting it to its own cluster, means that other ants lost it. This calls for a refining process in the next steps.

While transporting the records, the pheromone matrix A, is updated with the support of the utility rules in the clusters (lines 9–11).

For a predefined number of iterations, or until every ant becomes jobless (line 14), each ant proceeds according to the following logic: First it calculates the β_{ur} of its utility rule ur, (line 17) and checks if $\beta_{ur} < \beta_{predefined}$ or if there is less than k records embedding ur in its corresponding cluster, while there is still available respective records outside the cluster; in this case, function PickUp is called (lines 18-19).

At this point, the expert ant is responsible of bringing a record containing the utility rule in question, ur, to its representative cluster. Actually, the PickUp function chooses the cluster that has the highest density of ur (line 2) and picks a corresponding record r (line 7), withdraw it from its cluster (line 8) and drop it in the cluster representative of ur (line 9). Then, the pheromone matrix, A, is updated for both, the source and the destination clusters, based on what utility rules are found within the transported record r (lines 10–15). However, when the record r is picked from a cluster that has a size less then k; the expert ant finds a record from another cluster that can fill the gap of transporting r, by calling recursively the PickUp function; ensuring that the cluster preserves its size.

However, if $\beta_{ur} \geq \beta_{predefined}$, the expert ant is free to work for the benefit of another ant during the current iteration. In this case, the number of jobless ants increases (line 21) and function DropLoad is called (line 22) to find another ant to help. Function DropLoad finds the utility rule that still needs the most iterations to achieve the $\beta_{predefined}$ (line 2). Then, it calls the PickUp function to find a record that can be transported to the corresponding cluster.

At the end of the iterations, there exist u clusters, each representing mainly one utility rule. The resulting clusters may have sizes greater than the maximum cluster size δ allowed. To abide to the δ constraint of the disassociation technique, every cluster is passed to the Split function (line 28) and is split into smaller clusters having respectively a size less or equal to δ, when necessary.

The algorithm ends by vertically partitioning the resulting clusters from (*UDAC*) (line 30) and treats all the other records via the normal processes of the disassociation technique (line 31).

Algorithm 1 UDAC

```
Require: $\mathcal{T}$, $UR$, $k$, $\delta$, $\beta_{predefined}$, $it$
Ensure: $\mathcal{T}^*$
 1: $\mathcal{T}_{|UR} = \{r \mid r \in \mathcal{T}$ and $\exists\, ur \in UR$ and $ur \subseteq r\}$
 2: $u = |UR|$
 3: create $u$ clusters, $u$ ants and square matrix $A[u][u]$
 4: $it\_count = 0$
 5: while ($\mathcal{T}_{|UR} \neq \emptyset$) do
 6:     for each expert ant $a_i$ do
 7:         $C_i = C_i \cup r \mid r \in \mathcal{T}_{|UR}$ and $ur_i \subseteq r$
 8:         $\mathcal{T}_{|UR} = \mathcal{T}_{|UR} \setminus r$
 9:         for ($j = 0; j < u; j++$) do
10:             $A[i][j] = s(ur_i, C_j)$
11:         end for
12:     end for
13: end while
14: while ($it\_count < it$ or $jobless\_ants < u$) do
15:     $jobless\_ants = 0$
16:     for each expert ant $a_i$ do
17:         $\beta_{ur_i} = \frac{A[i][i]}{s(ur_i, \mathcal{T})}$
18:         if ($\beta_{ur_i} < \beta_{predefined}$ or $A[i][i] < k < s(ur_i, \mathcal{T})$) then
19:             PickUp($a_i, ur_i, 0$)
20:         else
21:             $jobless\_ants++$
22:             DropLoad($a_i$)
23:         end if
24:     end for
25:     $it\_count++$
26: end while
27: for each cluster $C_i$ do
28:     $ClustersSet = ClustersSet \uplus$ Split($C_i$)
29: end for
30: VerticalPartitioning($ClustersSet$)
31: Disassociation($\mathcal{T} \setminus \mathcal{T}_{|UR}$)
```

```
 1: procedure PickUp($a_i, ur_j, Recur_{count}$)
 2:     $C_{pu} = \arg\max(\frac{A[j][n]}{|C_n|}), \forall n! = j$
 3:     if $|C_{pu}| \leq k$ and $Recur_{count} < u$ then
 4:         $Recur_{count}++$
 5:         PickUp($a_i, ur_j, Recur_{count}$)
 6:     end if
 7:     $load(a_i) = r \mid r \in C_{pu}$ and $ur_j \subseteq r$
 8:     $C_{pu} = C_{pu} \setminus r$
 9:     $C_j = C_j \uplus load(a_i)$
10:     for ($l = 0; l < u; l++$) do
11:         if ($ur_l \subseteq load(a_i)$) then
12:             $A[l][pu]--$
13:             $A[l][j]++$
14:         end if
15:     end for
16: end procedure
```

```
1: procedure DropLoad($a_i$)
2:     $ur_d = argmax_{\forall j!=i}(s(ur_j, \mathcal{T}) * (\beta_{predefined} - \beta_{ur_j}))$
3:     PickUp($a_i, ur_d, 0$)
4: end procedure
```

```
 1: procedure Split($C_i$)
 2:     if $|C_i| > \delta$ then
 3:         create new cluster $C_{new}$
 4:         for (int $l = 0; l < \delta; l++$) do
 5:             $load(a_i) = r \mid r \in C_i$
 6:             $C_i = C_i \setminus r$
 7:             $C_{new} = C_{new} \uplus load(a_i)$
 8:         end for
 9:         $AntClusters = AntClusters \uplus C_{new}$
10:         Split($C_i$)
11:     else
12:         $AntClusters = AntClusters \uplus C_i$
13:     end if
14:     Return $AntClusters$
15: end procedure
```

5 EXPERIMENTS

5.1 Experimental settings

Table 3: Utility rules' characteristics

Utility Rules Set	set1	set2	set3	set4	set5	set6
Set Cardinality	5	10	15	20	25	30
Records Interdependency	1.35	1.39	1.38	1.39	1.65	1.82
Lowest Frequency	359	8	8	4	8	2
Highest Frequency	1204	85	1204	460	359	225
Records Count $\|\mathcal{T}_{\|UR}\|$	2379	231	2235	1173	898	823

In this section we present a set of experiments, evaluating our utility driven ant-based clustering technique, *UDAC*, in terms of privacy and utility. We choose for the experiments the *BMS*1 dataset, which contains click-stream E-commerce data, and 6 sets of utility rules extracted from the dataset with different characteristics shown in Table 3. Records Interdependency represents the average number of utility rules in the records of $\mathcal{T}_{|UR}$. We compare *UDAC* (with $\alpha = 0.8$) to the widely used unsupervised clustering technique, $k - means$, and horizontal partitioning (*HP*) of disassociation.

5.2 Utility privacy trade-off evaluation

In the disassociation technique, privacy of the data is preserved based on the $k^m - anonymity$ model through the process of vertical partitioning. Our work isn't challenged by redefining the vertical partitioning that is responsible of the privacy preservation. However, for any data analysis upon the anonymized data we should study what is the effect of the disassociation on the final output in terms of data accuracy. In the following experiments we highlight the effect of different clustering techniques on the final result of vertical partitioning, for the utility rules. On the same set of records representing the utility rules, we run respectively $k - means$ with the cosine distance and our *UDAC* algorithm, then we apply vertical partitioning from the disassociation technique for each cluster.

5.2.1 Relative loss.

The first metric we use to evaluate the loss in association with regard to the predefined utility rules, is the relative loss metric *RLM*, defined as follows:

$$RLM = \frac{\sum_{i=1}^{|UR|}(s(ur_i, \mathcal{T}) - s(ur_i, \mathcal{T}^*))}{\sum_{i=1}^{|UR|} s(ur_i, \mathcal{T})}$$

where, $s(ur_i, \mathcal{T})$ and $s(ur_i, \mathcal{T}^*)$ represent the support of the utility rule ur_i in the original dataset, $\mathcal{T}$, and in the anonymized dataset, $\mathcal{T}^*$ respectively.

Figure 3 shows a huge effect of the different clustering techniques on the loss of utility rules, with *UDAC* preserving the lowest *RLM* for all the sets compared to $k - means$ and *HP*. However, it is to be noticed that for the highly frequent set of utility rules, set1, the clustering techniques reveal near values for *RLM*. Since *HP* already works under the concept of items' frequency, it can preserve better utility for frequent itemsets, keeping it more challenging for the sets which contain infrequent utility rules like set2, which reveals this weakness of HP. These results manifest the crucial need for a special clustering technique for utility preservation, conform to the discussion in section 2.

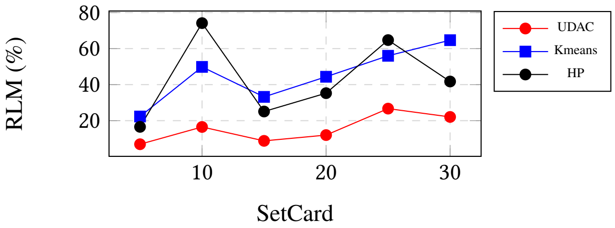


Figure 3: Relative Loss Metric

5.2.2 Average record partitioning. ARP is used to represent the average number of split-ups a record had to endure after vertical partitioning. We define *ARP* as:

$$ARP = \frac{\sum_{i=1}(|C_i| * count(R_{i_C}))}{|\mathcal{T}_{|UR}|}$$

where $count(R_{i_C})$ is the number of record chunks in cluster C_i and $|\mathcal{T}_{|UR}|$ is the number of records treated through special clustering. Low *ARP* means that less noise is added from vertical partitioning, being able to save more terms associated in the record chunks. Figure 4 shows an out-performance of *UDAC* over $k - means$ clustering. What is surprising is that although $k - means$ uses distance metrics to evaluate the similarity between the records over the whole set of items, while *UDAC* groups records based on the presence of utility rules in the records; the final vertical partitioning is better for *UDAC* than $k - means$s clustering. This means that beyond the items belonging to a utility rule, the other items present in $|\mathcal{T}_{|UR}|$ weren't vertically disassociated in an abusive way, reflecting efficiency of our solution in comparison to the classical clustering technique, $k - means$.

5.2.3 UDAC under the utility-privacy perspective. This experiment communicate the whole essence of the trade-off between utility and privacy in disassociation. We set $\beta_{predefined} = 0.8$, the maximum number of iterations = 3000 and run *UDAC* algorithm to see the average β_{ur} of the sets. To avoid the cluttering of the graphs, we only show, in Figure 5 the result for two sets: set1 in red and set6 in blue.

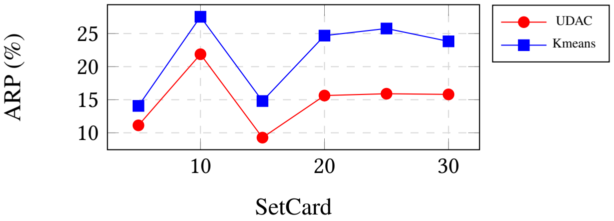


Figure 4: Average Record Partitioning

An average β_{ur} greater than $\beta_{predefined}$ is possible when the records' interdependency is low as in set1, this means that the representation of the utility rules in the clusters is more accurate. When, this interdependency between the records is higher, as in set6, it becomes harder to achieve the $\beta_{predefined}$. On another side, the graph shows a lower β_{ur} with the increase of the values of k and m, the higher the privacy is requested, the lower the utility can be achieved by safeguarding the utility rules without vertical partitioning.

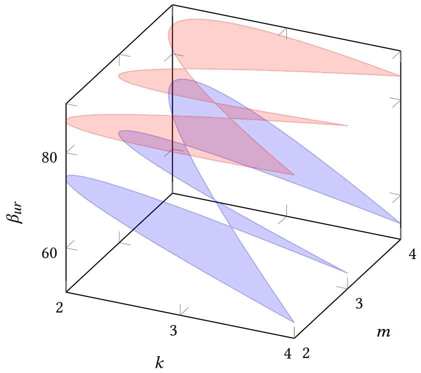


Figure 5: The k-m-β relationship

6 CONCLUSION

Choosing the right anonymization technique, depends mainly on the type of data in question and the desired result after anonymization. In fact, many techniques are proposed in the literature either for query answering or for publishing the data, while preserving the privacy of the users. Anonymization becomes harder when data must be analyzed after publishing it, and the challenge is to find a good trade-off between privacy and utility. Disassociation is a technique that provides a form of anonymization without altering the value of the data items. It guarantees $k^m - anonymity$ for associations within a cluster by vertically partitioning it into record chunks. In this paper, we analyze the loss of associations for aggregate analysis, showing its direct link to the clustering process of the records before vertical partitioning. Driven by this problem we propose the utility guided ant-based clustering algorithm, (*UGAC*), to drive the process of clustering for a set of records representing the predefined utility rules, to increase their utility by pushing their preservation non-partitioned in an indirect way. Finally, to test our algorithm, we compare our algorithm, for various properties of utility rules,

with the classical clustering technique, $k - means$, and the normal horizontal disassociation. The result shows that the information loss, for the utility rules clustered via our *UGAC* algorithm, decreases compared to the other two solutions.
From the theoretical analysis in section-2, a data analyst have to first calculate the average support of associations, to be able to execute analysis over a disassociated dataset. If the associations in questions aren't predefined and treated through *UGAC*, the accuracy of any analysis is under risk. In future works, we aim at publishing anonymized set-valued data, fully ready for analysis.

ACKNOWLEDGMENTS

This work is funded by the InMobiles company[1] and the Labex ACTION program (contract ANR-11-LABX-01-01). Computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté. This work is also jointly funded with support from the Hubert Curein CEDRE Programme number 40283YK and the National Council for Scientific Research in Lebanon CNRS-L and the Antonine University.

REFERENCES

[1] N. Awad, B. A. Bouna, J.-F. Couchot, and L. Philippe. Safe Disassociation of Set-Valued Datasets. *arXiv e-prints*, page arXiv:1904.03112, Apr 2019.

[2] S. Barakat, B. al Bouna, M. Nassar, and C. Guyeux. On the evaluation of the privacy breach in disassociated set-valued datasets. In C. Callegari, M. van Sinderen, P. G. Sarigiannidis, P. Samarati, E. Cabello, P. Lorenz, and M. S. Obaidat, editors, *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRYPT, Lisbon, Portugal, July 26-28, 2016.*, pages 318–326. SciTePress, 2016.

[3] J. C. Bezdek, R. Ehrlich, and W. Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203, 1984.

[4] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, G. Livraga, S. Paraboschi, and P. Samarati. Extending loose associations to multiple fragments. In *Proceedings of the 27th International Conference on Data and Applications Security and Privacy XXVII*, DBSec'13, pages 1–16, Berlin, Heidelberg, 2013. Springer-Verlag.

[5] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 356–363, 1991.

[6] M. Dorigo and M. Birattari. *Ant colony optimization*. Springer, 2010.

[7] M. Dorigo and L. M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66, 1997.

[8] C. Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340, 2011.

[9] A. M. Fard and K. Wang. An effective clustering approach to web query log anonymization. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–11. IEEE, 2010.

[10] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, and Z. Tang. Ant colony optimization with clustering for solving the dynamic location routing problem. *Applied Mathematics and Computation*, 285:149–173, 2016.

[11] Y. He and J. F. Naughton. Anonymization of set-valued data via top-down, local generalization. *Proc. VLDB Endow.*, 2(1):934–945, Aug. 2009.

[12] D. Karaboga and C. Ozturk. A novel clustering approach: Artificial bee colony (abc) algorithm. *Applied soft computing*, 11(1):652–657, 2011.

[13] J. Kennedy. Particle swarm optimization. *Encyclopedia of machine learning*, pages 760–766, 2010.

[14] N. Labroche, N. Monmarché, and G. Venturini. A new clustering algorithm based on the chemical recognition system of ants. In *ECAI*, pages 345–349, 2002.

[15] T. Li, N. Li, J. Zhang, and I. Molloy. Slicing: A new approach for privacy preserving data publishing. *IEEE Trans. Knowl. Data Eng.*, 24(3):561–574, 2012.

[16] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[17] P. Moradi and M. Rostami. Integration of graph clustering with ant colony optimization for feature selection. *Knowledge-Based Systems*, 84:144–161, 2015.

[18] Z. Sadeghi and M. Teshnehlab. Ant colony clustering by expert ants. In *2008 11th International Conference on Computer and Information Technology*, pages 94–100. IEEE, 2008.

[19] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6):1010–1027, 2001.

[20] P. Shelokar, V. K. Jayaraman, and B. D. Kulkarni. An ant colony approach for clustering. *Analytica Chimica Acta*, 509(2):187–195, 2004.

[21] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[22] M. Terrovitis, N. Mamoulis, J. Liagouris, and S. Skiadopoulos. Privacy preservation by disassociation. *Proc. VLDB Endow.*, 5(10):944–955, June 2012.

[23] A. Vattani. The hardness of k-means clustering in the plane. *Manuscript, accessible at http://cseweb. ucsd. edu/avattani/papers/kmeans_hardness. pdf*, 617, 2009.

[24] J. Wang, C. Deng, and X. Li. Two privacy-preserving approaches for publishing transactional data streams. *IEEE Access*, pages 1–1, 2018.

[25] K. Wang, P. Wang, A. W. Fu, and R. C.-W. Wong. Generalized bucketization scheme for flexible privacy settings. *Information Sciences*, 348:377–393, 2016.

[26] Z.-H. Wang, C.-C. Chang, and M.-C. Li. Optimizing least-significant-bit substitution using cat swarm optimization strategy. *Information Sciences*, 192:98–108, 2012.

[27] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *Proceedings of 32nd International Conference on Very Large Data Bases (VLDB 2006)*, Seoul, Korea, Sept. 12-15 2006.

[28] H. Zhu, Y. Wang, K. Wang, and Y. Chen. Particle swarm optimization (pso) for the constrained portfolio optimization problem. *Expert Systems with Applications*, 38(8):10161–10169, 2011.

[1] www.inmobiles.net

EQL-CE: An Event Query Language for Connected Environments

Elio Mansour
Univ Pau & Pays Adour
E2S UPPA, LIUPPA
Mont-de-Marsan, 40000, France
elio.mansour@univ-pau.fr

Richard Chbeir
Univ Pau & Pays Adour
E2S UPPA, LIUPPA
Anglet, 64600, France
richard.chbeir@univ-pau.fr

Philippe Arnould
Univ Pau & Pays Adour
E2S UPPA, LIUPPA
Mont-de-Marsan, 40000, France
philippe.arnould@univ-pau.fr

ABSTRACT

Recent advances in sensor technology and information processing have allowed connected environments to impact various application domains. In order to detect events in these environments, existing works rely on the sensed data. However, these works are not re-usable since they statically define the targeted events (i.e., the definitions are hard to modify when needed). Here, we present a generic framework for event detection composed of (i) a representation of the environment; (ii) an event detection mechanism; and (iii) an Event Query Language (EQL) for user/framework interaction. This paper focuses on detailing the EQL which allows the definition of the data model components, handles instances of each component, protects the security/privacy of data/users, and defines/detects events. We also propose a query optimizer in order to handle the dynamicity of the environment and spatial/temporal constraints. We finally illustrate the EQL and conclude the paper with some future works.

CCS CONCEPTS

• General and reference → General conference proceedings; **• Information systems → Query representation**; **• Theory of computation → Grammars and context-free languages**; **• Computer systems organization → Sensor networks**.

KEYWORDS

Event Query Language, Internet of Things, Sensor Networks

ACM Reference Format:
Elio Mansour, Richard Chbeir, and Philippe Arnould. 2019. EQL-CE: An Event Query Language for Connected Environments. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331076.3331103

1 INTRODUCTION

Recent advances in the fields of Information & Communication Technologies (ICT), Big Data, Sensing Technologies, and the Internet of Things (IoT) have paved the way for the rise of smart connected environments. These environments are defined as infrastructures that host a network of sensors capable of providing data that can be later mined and processed using advanced techniques, for high level applications. Hence, Sensor Networks (SN) are currently impacting numerous domains (e.g., medical, environmental, cities, buildings). This allowed a plethora of sensor-based applications such as monitoring a patient's health [20], detecting fires in the wilderness [24], monitoring pollution levels or traffic congestion in a city[15], and optimizing energy consumption/occupants' comfort in buildings[1, 8, 14, 19, 23, 25]. Even though these applications have different objectives, they all rely on sensed data from the environment in order to detect specific events (e.g., a stroke for a patient, a volcanic eruption, a storm, polluted air in a city, temperature rising in an office). Therefore, these applications share the following needs: (i) representing the infrastructure and the sensor network of the connected environment; (ii) defining and detecting the targeted events; and (iii) protecting the security of the sensed data and the privacy of the users in the environment (e.g., protecting patients' medical records). In the aforementioned works, the authors do not emphasize on the environment's representation and define the events statically. They also proposed event detection mechanisms that perfectly fit the description of the targeted events. This is constraining since these works are not re-usable in different contexts. Event Query Languages (EQL) have been proposed to overcome this issue. Users express their needs through EQLs by defining the structure of the targeted events. However, existing languages [2, 3, 6, 7, 9–11] focus mainly on the event descriptions and do not consider other environment components (e.g., infrastructure, sensor network, application domain). They share the following limitations:

(1) Lack of *considered components*. It is important that the EQL allows the definition of the entire connected environment. This includes components related to the environment itself, its sensor network, the targeted events, and the application domain.
(2) Lack of *considered functionality*. It is important that the EQL (i) allows the definition of components (e.g., buildings, sensors, data, events); (ii) allows the manipulation of component instances (e.g., inserting new instances, updating, deleting, selecting them); and (iii) protects the security/privacy of data/users.
(3) Lack of *re-usability*. It is important that the EQL remains generic and independent from any technological constraints or underlying infrastructure. Some languages heavily rely on a specific syntax or data model (e.g., SQL-based, or SPARQL-based) and this limits their re-usability in different contexts.

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331103

In order to consider the dynamicity of connected environments and spatial/temporal distributions, we consider the following limitations as well. First, the difficulty in *handling dynamic environments.* Since sensors might breakdown, or mobile sensors could change locations or even enter/exit the network, sensors/observations that are needed for a previously defined event might become unavailable. Therefore, it is important that the EQL allows query re-writing in order to update obsolete event definitions. This entails replacing missing sensors by others capable of providing the required data or replacing missing observations with others that fit the event definition. Second, the lack of *spatial distribution* of sensors. Since the sensors' locations impact event detection, the EQL should allow users to define spatial distributions of the sensors over the infrastructure in order to better detect the targeted events. This entails specifying where each sensor should be located or how they should be distributed over the space (e.g., nearest sensors to a point of interest, sensors within a range of a point of interest, sensors that fit a mathematical distribution around a point of interest). Finally, the lack of *temporal distribution* of sensor observations. Since sensors provide observations at specific rates, one could end up with either: (i) big volumes of unnecessary data (if the rate is too quick); or (ii) undetected events (if the rate is too slow). Therefore, it is important to have an EQL that allows the adjustment of the temporal distribution of sensor observations based on events' needs/requirements. This entails specifying which sensor observations/sensing rates are considered for a specific event, or selecting a temporal distribution of these observations (e.g., the closest observations to a certain point in time, all observations within a temporal range, distributed sensing rates).

Many other challenges emerge when considering an EQL for connected environments (e.g., handling big volumes of data, continuous heterogeneous data streams). However, in this paper, we focus mainly on the aforementioned limitations. Hence, we propose here an EQL specifically designed for connected environments and partitioned into three layers: conceptual, logical, and physical. It allows (i) the composition of high level generic queries that can be parsed into various data model specific languages (re-usability); and (ii) full coverage of components and functionality (we will detail security related tasks in a dedicated future work). We also propose a query optimizer module that will handle spatial/temporal distributions and query re-writing in order to redefine components that need to evolve when handling the environment's dynamicity (the optimizer will be fully detailed in a separate work). Our proposal, denoted EQL-CE, is part of a global framework for event detection in connected environments which we will also present in this work.

The remainder of the paper is organized as follows. Section 2 presents a scenario that motivates our proposal. Section 3 evaluates existing approaches. Section 4 presents our event detection framework and details EQL-CE. An illustration example is presented in Section 5. Finally, Section 6 concludes the paper and discusses future research directions.

2 MOTIVATING SCENARIO

In order to motivate our proposal, consider the following scenario that illustrates a smart mall. This is a simplified example that illustrates the setup, the needs, and motivations behind our proposal. Of course, it does not summarize all needs found in a connected environment/event detection application scenario. Figure 1 details the infrastructure's location map, and individual locations (i.e., shops and open areas). The mall is equipped with a hybrid sensor network having static/mobile sensors, single sensor nodes/multi-sensor devices capable of monitoring the environment and producing scalar/-multimedia observations (e.g., temperature, video). A manager uses an Event Query Language (EQL) in order to define/detect interesting events that occur within the mall's premises. Although this seems enough to manage the smart mall, many improvements can still be integrated:

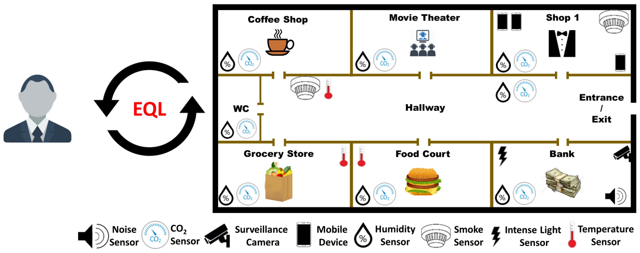


Figure 1: The Smart Mall

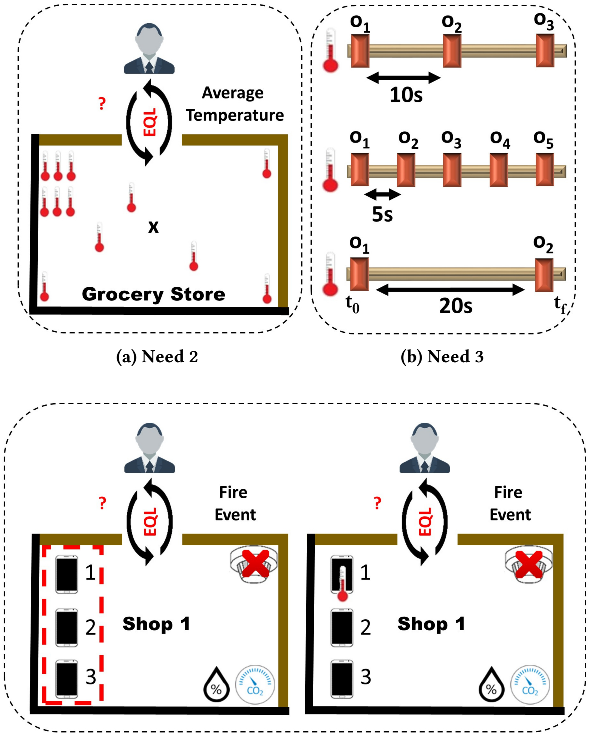
- Need 1: Modeling the environment and its sensor network. Before defining and detecting events, a mall manager needs to represent the smart mall using the EQL. This includes defining the infrastructure (i.e., the mall), the locations (e.g., shops), and their spatial relations. Then, the manager needs to define the sensor network that is hosted in the mall. This entails modeling the available sensors (e.g., temperature, humidity), their deployment locations, the data they sense and so on. Once all component structures are defined, the mall manager needs to use the EQL to create instances of each component (e.g., temperature sensor in food court). This is currently not possible since the EQL used in the example only defines events.
- Need 2: Measuring the average temperature in the grocery store (for food storage concerns). The mall manager uses the existing EQL to define the targeted event (i.e., the average temperature in the grocery store). The EQL allows the manager to consider all sensors within the area of interest. However, Figure 2.a shows that the sensors are not evenly distributed in the store (most are located in the upper left corner). Hence, considering all sensors and calculating the average will produce a biased temperature measure that does not reflect the reality of the situation. This can be solved by allowing the manager to define a specific distribution of sensors over the space (e.g., even distribution, only considering sensors within a range of the center of the store). The current setup is limited since it does not allow the definition of spatial distributions of sensors.
- Need 3: Minimize data overload/missed events. Currently, the manager can use the EQL to define one sensing rate for all sensors or sensor types (e.g., temperature, humidity). This is constraining since (i) a quick sensing rate overloads the system with big volumes of unwanted/unnecessary data;

and (ii) a slow sensing rate could lead to missing events that began and ended in a short time lapse. Therefore, the temporal distribution of sensor observations (i.e., a start time, a specific rate, a stop time) should be based on the event definition and therefore considered/handled in the event queries (e.g., selecting the closest observations to a time of interest, considering different sensing rates from various sensors at once). The EQL used by the mall manager does not allow such customization of temporal constraints (cf. Figure 2.b).

- Need 4: Detecting a fire event in Shop 1. The mall manager defines this fire event using the EQL. His/Her definition relies on the smoke, humidity, and CO_2 sensors located in Shop 1. However, what if the smoke sensor broke down ? Or what if the mobile device that he/she was depending on left the shop ? Then, the previously defined event query will become obsolete since there are no more smoke observations coming from shop 1, and there is no way of changing the event definition. Hence, query re-writing is necessary in order to update the definition: (i) by replacing the smoke sensor by another capable of providing the same data (e.g., mobile device 1 - cf. Figure 2.c.left); or (ii) by replacing the event describing feature smoke by another (e.g., temperature from mobile device 1 if no other sensors can provide smoke observations - cf. Figure 2.c.right). The current EQL is limited since it does not allow such re-writing.

In order to address the aforementioned needs, the EQL should provide a means for defining the structure of various components related to the environment, sensor network, targeted events, and application domain. Moreover, the EQL should not be limited to defining components. Its functionality should extend to managing instances of these components, and protecting the security/privacy of the data/users (cf. Need 1). In addition, customizing the sensors' spatial distribution over the infrastructure/environment based on event requirements is required (cf. Need 2). This benefits the event detection since it provides the user with the ability to customize the setup in the way that he/she believes is optimal. The same is also applied for temporal distribution of sensor observations. The EQL should allow the user to select specific observations, or a set of distributed observations in time (e.g., considering different sensing rates, temporal distance to a point in time) when defining the event (cf. Need 3). Finally, the EQL should allow re-writing queries (cf. Need 4) to handle the dynamicity of the connected environment. This is especially beneficial when faults or sensor breakdowns/mobility can render some event definitions obsolete. However, when considering various components, functionality, data distribution (e.g., spatial, temporal), and query re-writing the following challenges emerge:

- Challenge 1: How to model components and inter-component relations? How to establish ties between the different connected environment elements (i.e., environment, sensor network, events, and application domain)?
- Challenge 2: How to define different query types to cover all the required functionality?



(a) Need 2

(b) Need 3

(c) Need 4

Figure 2: Spatial Distribution (a) | Temporal Distribution (b) | Query Re-writing (c)

- Challenge 3: How to establish a generic query syntax that can be re-used regardless of the underlying infrastructure (e.g., in a traditional database or in an ontology data model)?
- Challenge 4: How to integrate variables that specify spatial/temporal distributions in the query syntax ? How to propose different types of distribution queries ?
- Challenge 5: How to enable query re-writing upon user request ? How to replace missing sensors/event describing features when re-writing a query ?

Therefore, we propose here a high-level generic event query language, denoted EQL-CE, capable of covering all components. Our covered functionality are partitioned into three main categories for component definition, manipulation of component instances, and data protection (to be discussed in a future work). We also propose a query optimizer that handles query re-writing and spatial/temporal distribution functions. In this paper, we present the optimizer but leave the details of the query re-writing and distribution functions to a separate dedicated work.

3 RELATED WORK

In this section, we review existing works on Event Query Languages (EQL). We propose the following criteria based on the challenges and limitations discussed in Section 2:

- Criterion 1. Component/Functionality Coverage: Denoting if the EQL covers (i) the entire components that constitute a connected environment (i.e., environment, sensor network, application domain, and event related components); and (ii) the entire set of functionality needed for the definition of components, the manipulation of their instances, and protection of the data/user security and privacy (cf. Need 1).
- Criterion 2. Re-usability: Indicating if the EQL is generic and technology independent in order to re-use it in various setups with different underlying infrastructures (e.g., traditional database, ontology). It is beneficial to have a high level, generic, and declarative EQL that can be parsed into data-model specific languages (instances). This facilitates its integration in various contexts.
- Criterion 3. Spatial/Temporal Distributions: Specifying if the EQL allows (i) spatial distribution queries (e.g., selecting sensors that are distributed based on a mathematical law, within a specific range, or near a point of interest); and (ii) temporal distribution queries (e.g., selecting sensor observations that are closest to a point in time that have various sensing rates). This is important for the definition of specific events where such level of detail is required (cf. Needs 2 and 3).
- Criterion 4. Handing Environment Dynamicity: Stating if the EQL provides the means to modify the structure of previously defined components (e.g., events) in order to keep up with environment changes. This is useful in a dynamic setup, where sensor mobility causes gain/loss of data in certain areas of the environment (cf. Need 4).

We group the existing works into three main categories: (i) conceptual languages (e.g., Event-Condition-Action languages) ; (ii) logical languages; and physical languages (e.g., SQL/SPARQL-based languages). We compare in the following some works from each category (we do not detail here every existing event query language for the sake of brevity).

3.1 Conceptual Languages

This category of languages includes Event Condition Action (ECA) languages that allow the declaration of three event attributes: (i) an event name or label; (ii) a set of conditions (the pattern) that best define the event; and (iii) the set of actions that should be triggered once the event is detected. In [9], the authors propose an intuitive event query language denoted SNOOP. They follow the ECA model when defining event structures. They integrate operators for inter-condition relations (e.g., conjunction, dis-junction, and sequence) and represent repetitive events through the usage of the periodic/aperiodic operators. In [6], the authors propose a language denoted CeDR. In comparison with SNOOP, CeDR adds a WHERE clause for filtering statements and has a wider range of operators. Therefore, CeDR is considered more expressive in terms of event pattern description. CeDR also includes an event lifetime operator and a detection window operator. The authors in [11] propose an event query language for data streams called SaSE. They include the WITHIN and RETURN statements to respectively declare sliding time windows and the required output. SaSE also allows event pattern operators (similar to CeDR) in a WHERE clause.

Discussion: The aforementioned works are intuitive, practical, and allow various composition operators for event definition. Their syntax is also independent from specific data models (e.g., SQL or SPARQL). However, they all suffer from the same limitations. None of them covers the environment or sensor network definition in their queries (cf. Criterion 1 - Component Coverage). They mainly focus on the definition and retrieval of events while neglecting other tasks such as updating definitions or inserting data (cf. Criterion 1 - Functionality Coverage). They also do not consider spatial/temporal distributions (cf. Criterion 3).

3.2 Logical Languages

This category includes works that define events in logic style formulas. To give a few examples, consider ETALIS[3]. This EQL describes events as rules. The authors propose a set of temporal relations and composition operators to define the event patterns. The syntax of the rules is independent of any underlying data model. XChangeEQ[7] is another logical language. The authors allow the following features in its queries: (i) data-related operations such as variable bindings and conditions containing arithmetic operations; (ii) event composition operators such as conjunction, dis-junction, and order; (iii) temporal and causal relations between events in the queries; and (iv) event accumulation, for instance aggregating data from previous events to discover new ones.

Discussion: The aforementioned languages are re-usable in different contexts since their syntax, a logical rule-based notation, is independent of specific data models (cf. Criterion 2). They also cover the majority of temporal and composition operators. However, they do not cover spatial/temporal distributions (cf. Criterion 3). These languages have not fully detailed query re-writing (cf. Criterion 4), and they mainly focus on the events. They cannot be used to define and manage the environment and sensor network components (cf. Criterion 1).

3.3 Physical Languages

This category includes data model specific languages. We detail here languages that were specifically designed for either relational database or linked data management systems. Therefore, the following EQLs are either inspired from or directly extend SQL/SPARQL. ESPER[10] is an implementation for event detection in database systems. The authors proposed an SQL-like syntax for event processing. Therefore, known operators such as CREATE, SELECT, INSERT, UPDATE, and DELETE are available for event definition and detection. ESPER also includes temporal operators and a specific statement for event definition (i.e., the pattern). In addition to the aforementioned advantages, this language has a fast learning curve since it is highly similar to traditional SQL. CQL[4] is another language that can be used for event definition/retrieval. CQL extends SQL by emphasizing on continuous data streams/queries. The authors add temporal operators, sliding windows, and window parameters to better handle continuous data. Many languages extend SPARQL for linked data management systems. For instance, C-SPARQL[5] extends SPARQL to consider stream data in the queries. To do so, the authors integrate sliding time windows. SPARQL-ST[17] extends SPARQL by adding operators for spatial/temporal queries. This covers the definition and manipulation of spatial shapes and temporal entities. Finally, EP-SPARQL[2] integrates event processing operators (e.g., sequence) into the SPARQL syntax. This work allows the definition of simple and complex event patterns in a linked data management system.

Discussion: The aforementioned works are all user friendly since they extend known languages. They cover definition and manipulation queries for various components or entities (cf. Criterion 1). They also provide a basis for spatial/temporal operators and query re-writing. However, distribution queries are not considered (cf. Criterion 3) and their high reliance on a specific data model syntax (SQL or SPARQL) limits their re-usability in different systems (cf. Criterion 2). For instance, EP-SPARQL cannot be used in a relational database infrastructure.

To conclude this section, none of the mentioned works fully considers our entire list of criteria. Therefore, we propose in the following section the Event Query Language for connected environments (EQL-CE). Our proposal has three layers (conceptual, logical, and physical). It ensures re-usability, handles dynamic environments, fully covers the components/required functionality, and integrates spatial/temporal distribution variables in its queries.

4 EQL-CE: AN EQL FOR CONNECTED ENVIRONMENTS

In order to highlight the usage of EQL-CE, we present here an overview of our framework for event detection in connected environments. This framework includes the following modules: (i) a data model representing the connected environment; (ii) an event Virtual Machine (eVM) for event detection; and (iii) an event query language for user/framework interaction. We start by briefly describing these modules. Then, we detail our proposed event query language for connected environments, denoted EQL-CE.

4.1 Event Detection Framework

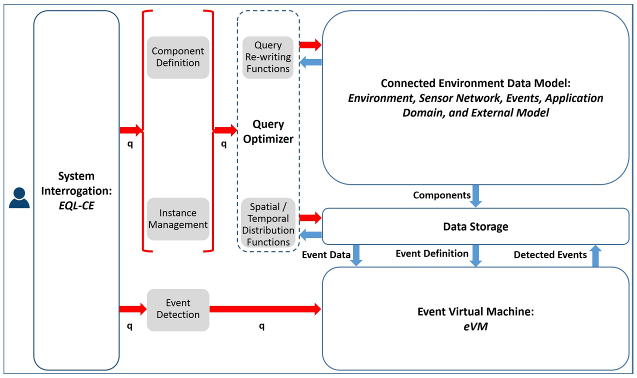

System Interrogation: EQL-CE
q
Component Definition
Instance Management
q
Query Re-writing Functions
Query Optimizer
Spatial / Temporal Distribution Functions
Connected Environment Data Model: Environment, Sensor Network, Events, Application Domain, and External Model
Components
Data Storage
Event Data
Event Definition
Detected Events
Event Detection
q
q
Event Virtual Machine: eVM

Figure 3: Global Framework Overview

Figure 3 illustrates our event detection framework. It contains three main modules:

- An event query language for connected environments (EQL-CE) and its query optimizer.
- A data model for connected environment representation.
- An event Virtual Machine for event detection (eVM).

4.1.1 Event Query Language. Users interrogate the system using the event query language. It is pivotal since it affects both the data model and the event Virtual Machine (event detector) modules. EQL-CE offers queries that can be used to define the structures of the data model components (i.e., entities that represent the connected environment). In addition, the language allows users to import external data models in the framework. Once the data model is defined, it is saved in the data storage. EQL-CE also manages instances of the previously defined components. It supports operations such as inserting new instances or even modifying, deleting, and retrieving existing ones. Also, the security and privacy of data/users can also be provided by EQL-CE via specific queries. From an event detection standpoint, users can trigger the event Virtual Machine via the query language in order to detect specific events. Finally, the query optimizer allows re-writing queries when needed, and can integrate spatial/temporal distribution functions in the queries (cf. Criterion 3 and 4). Both EQL-CE and its query optimizer will be further detailed in the following subsection.

4.1.2 Data Model. The data model of the connected environment gathers components that describe the environment itself, the sensor network, the events, and the application domain. When considering the environment, one might represent physical, real world, infrastructures such as buildings or offices and all their characteristics. This includes spatial descriptors (e.g., location maps, zones, individual locations, spatial relations), and specific entities that can be found in the environment (e.g., machines, equipment, devices). When considering the sensor network, one might represent sensors, observable properties, scalar/multimedia data, and so on. The targeted events should also be defined and described in the model. This includes event features, types, and patterns. Finally, the application domain is also a part of the model since it affects both the events and the environment. For instance medical events (e.g., high heart rate) differ from environmental events (e.g., temperature overheat in a room). Similarly, the equipment and entities found in a mall are different from the ones found in a hospital.

4.1.3 Event Detector. We proposed the event Virtual Machine (eVM) in a previous work [16]. It is an event detector that needs an event definition and a set of data objects (e.g., sensor observations) in order to detect targeted events. eVM is re-usable in different contexts, extensible, accepts various datatypes, easy to integrate, and requires low human intervention. The event detection process starts by retrieving the targeted event definition form the storage unit. The event definition is analyzed first in order to check its describing features. For instance a fire event is described by the following features: time, location, temperature, smoke, and CO_2. Then, the pre-processor retrieves data objects (e.g., sensor observations) having attributes related to these features (e.g., smoke, temperature, CO_2 observations). Once this is done, we use Formal Concept Analysis (FCA), a conceptual clustering technique, in order to construct a graph from the selected data objects/attributes. Finally, we detect the targeted events by examining the graph nodes and selecting the ones that are compatible with the event definition. Also, eVM is pluggable in the framework and can be replaced by any other event detector that requires data and an event definition in order to detect events. We do not detail the event detection mechanism in this paper since the aim is to focus on the event query language for connected environments.

4.2 The EQL-CE Proposal

We structure our proposal into three layers: (i) the conceptual layer provides an overview of the connected environment's components and their relations in the form of a graph; (ii) the logical layer allows the construction of generic queries written in EBNF (Extended Backus Norm Form) syntax; and (iii) the physical layer parses the EBNF queries into a data model-specific language (e.g., SQL, SPARQL) and executes the parsed queries. A simplified overview of EQL-CE is presented in Figure 4. In the following we detail each layer separately.

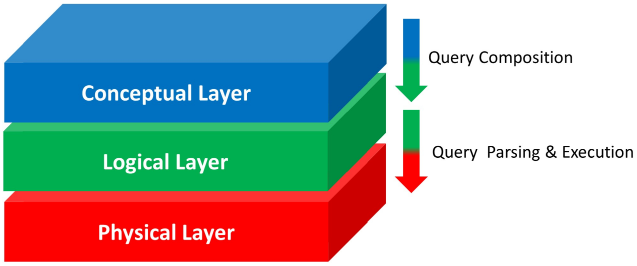


Figure 4: EQL-CE Overview

4.2.1 ***Conceptual Layer****.* Here, we detail the top layer of EQL-CE. The aim is to provide a clear and easy to exploit conceptual view of the connected environment. Therefore, we use a graph to represent the various elements (i.e., components and properties). The latter are split into the following categories (cf. Figure 5):

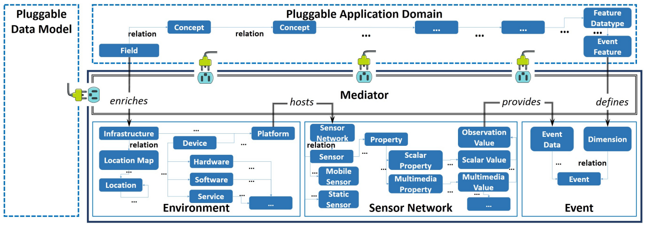


Figure 5: EQL-CE Conceptual Layer

Core Modeling: This part contains the basic elements that always exist in a connected environment. For a clear organization, we group the elements into the following two parts:

- Sensor Network modeling, where we represent (i) *sensor networks*; (ii) various *sensor* types (e.g., static, mobile); (iii) the different types of *properties* (i.e., scalar, multimedia) observed by sensors; and (iv) the *observation values* produced by sensors (i.e., textual values, multimedia objects and their respective metadata).
- Environment modeling, where we represent (i) *platforms* (i.e., infrastructure, devices) that host sensors or sensor networks; (ii) physical *infrastructures*, such as buildings, and their detailed description (i.e., *location maps*, spatial relations); (iii) *devices*, such as mobile phones, and their detailed description (i.e., *hardware*, *software*, provided *services*).

Many other components can still be added to the core part. The full description of the environment and sensor network can be inspired from ontologies such as SOSA/SSN [12] and HSSN (Hybrid Semantic Sensor Network).[1]

Event Modeling: This part contains the representation of events that one might wish to detect in a connected environment. Here, the application domain should also be considered since it affects the definition of specific events. For instance a body overheating (medical) event cannot be defined the same way as a room overheating (environmental) event. Hence, the application domain dictates the type of an event, its describing features, its pattern, and the required data for its detection. Therefore, we do not detail the event modeling, we keep it generic and restrict it to the following components: (i) *event* that defines an event and its type; (ii) *dimensions* to mathematically represent the event features (provided by the Application Domain) in a n-dimensional space; and (iii) *event data* to represent sensor observations that contributed in each event. This allows us to have a generic event definition that applies to various events from different application domains. All context specific details are defined in the application domain and then imported in the event definition via the mediator.

Application Domain Modeling: This part represents the application domain (e.g., medical, energy, military). Since these elements differ from one field to another, this part is pluggable into the conceptual model. It contains basic components/properties denoted *concepts* and *relations* respectively. Instances of the *concept* component can be used to define any domain specific entities, and instances of the *relation* property can be used to interconnect the *concepts* (e.g., Figure 5 shows an *Event Feature* concept that helps define event *dimensions*). This allows the customization of environment descriptions and event definitions based on specific contexts. For instance, one might wish to represent medical equipment and health related constraints when modeling a hospital environment. These elements are not the same when describing a shopping center. Similarly, what describes medical related events is different from normal every day events that happen in a mall. To conclude, this part of the data model complements the event description on one side, and enriches the environment representation on the other.

The Mediator: This part of the conceptual model only contains properties that ensure the interconnection of the previously mentioned parts (i.e., the core, event, and application domain). For instance, a *platform hosts a sensor network*, the *observation values* produced by the sensors *provide event data*, the event *dimensions* are defined by *event features*, and the concept *field enriches* the description of an *infrastructure* based on the application domain. In addition, the mediator can also be used to plug in an external data model and align it with the existing elements.

4.2.2 ***Logical Layer****.* The middle layer of EQL-CE, denoted the logical layer, allows users to compose/design their queries. The

[1] http://spider.sigappfr.org/research-projects/hybrid-ssn-ontology/

process starts by choosing a specific query type. To cover a wider set of functionality (cf. Criterion 1), we provide three main groups of queries:

- The Component Definition Language defines the structure of components. Various query types are included in this group (e.g., CREATE, ALTER, RENAME, DROP).
- The Component Manipulation Language handles component instances. Here we propose the following query types: SELECT, INSERT, UPDATE, and DELETE.
- Component Access Control (e.g., GRANT, REVOKE). These queries manage access rights to component data. We detail access control tasks in a dedicated future work.

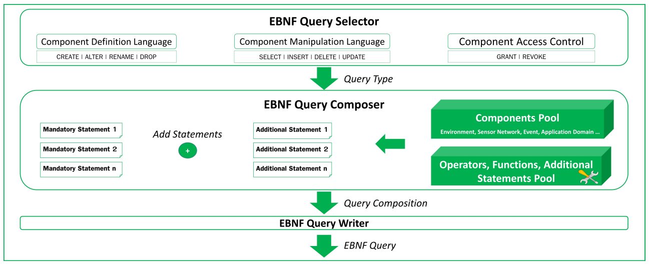


Figure 6: EQL-CE Logical Layer

The process of composing a query is described in Figure 6. First, the user chooses the query type (e.g., CREATE, INSERT, DELETE). Then, the user starts filling the mandatory statements (e.g., what to CREATE, what to SELECT, from which component). Once this is done, the user can add optional statements for filtering, ordering, calling external functions. Finally, the query is written using an Extended Backus-Naur Form syntax, denoted EBNF [22]. This context-free grammar is used to formally describe programming languages. It extends the Backus-Naur Form (BNF). We use EBNF since it allows the conception of technology independent queries (i.e., queries that do not depend from any data model specific syntax). This highlights the ability to re-use (cf. Criterion 2) EQL-CE in different setups, since EBNF can later be parsed, in the physical layer, to a specific data model instance, such as SQL or SPARQL, depending on the underlying infrastructure [13, 18, 21]. Any component from the conceptual model (i.e., related to the environment, sensor network, event, and application domain modeling) can be defined, manipulated, and protected using these queries (cf. Criterion 1). Finally, the EBNF query is sent to the physical layer.

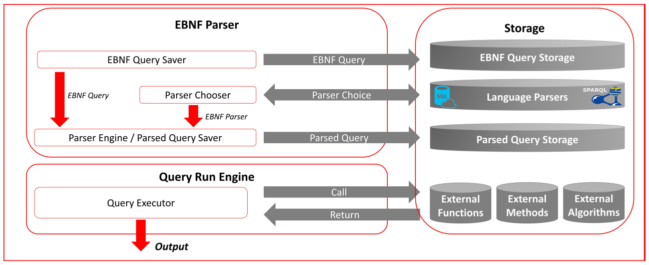


Figure 7: EQL-CE Physical Layer

4.2.3 ***Physical Layer & Query Optimizer.*** The bottom layer of EQL-CE (cf. Figure 7) saves the received EBNF queries in a dedicated storage unit for future use. Then, it parses the aforementioned queries into a specific syntax depending on the underlying data model (e.g., SQL, SPARQL). Finally, the parsed query is saved and sent to the query run engine where it is executed. If needed, external functions, methods, or even algorithms are called (e.g., string comparison functions, mathematical libraries). All the above describes how EQL-CE can be re-used in various contexts, since it is independent from any technological infrastructure (cf. Challenge 3). Using the EBNF queries, one can define the data model and all its various related components (cf. Challenge 1). In addition, EQL-CE allows users to handle instances of each component for data retrieval, modification, deletion, security/privacy, and event detection by providing a plethora of functionality (cf. Challenge 2). However, when defining specific events, one might need to manage the spatial distribution of sensors over a location (cf. Need 2). For instance, consider k-nearest sensors to a specific location, or all sensors within a range R of a point in space. Also, one might consider mathematical distributions of sensors over a zone (e.g., even distribution). Similarly, one might need to manage the temporal distribution of sensor observations for specific events (cf. Need 3). For example, selecting the k-most recent sensor observations, or all observations that were produced during a specific time lapse. Also, one might need to select observations based on specific sensing rates. To do so, the query optimizer allows the integration of spatial/temporal distribution functions in the queries (cf. Criterion 3 and Challenge 4). Finally, in dynamic connected environments sensors might suffer from breakdowns, mobile sensors could enter/leave the network, or even change locations. This is challenging since event definitions rely on sensors and their provided observations. Hence, some previously defined events might become obsolete over time. Therefore, in some cases, queries need to be re-written or updated in order to handle the dynamicity of the environment, and keep up with its evolution (cf. Criterion 4 and Challenge 5). This is also possible via the query optimizer. In this paper, we do not fully detail the query re-writing and spatial/temporal distribution functions. We leave this to a dedicated future work.

5 ILLUSTRATION EXAMPLE

In this section, we illustrate how EQL-CE works. The aim here, is to demonstrate the component syntax and provide some EBNF query examples. To do so, we consider the smart mall scenario of Section 2. For the sake of brevity, we do not define the entire connected environment (e.g., all locations, sensors in the mall). A fully detailed example (i.e., containing various query types and components) can be found on the following link: http://spider.sigappfr.org/research-projects/eql-ce/smart-mall/.

5.1 Environment Modeling

The mall is an infrastructure having a location map and various locations (e.g., shop 1, food court) that are tied by spatial relations. First, we need to define these components. Then, we INSERT instances. Syntax 1 defines an infrastructure as an entity that has a location map, and a set of embedded platforms (e.g., infrastructures, devices). A location map contains various locations (cf. Syntax 2).

Finally, each location can be spatially tied to other locations (cf. Syntax 3).

Syntax 1: Defining the structure of an Infrastructure

```
CREATE INFRASTRUCTURE ( <id> = <string> ,
[ LOCATION MAP <id>  = <string> ,] [ { HOSTED PLATFORM <id>  =  <string> } ] ) ;
```

Syntax 2: Defining the structure of a Location Map

```
CREATE LOCATION MAP ( <id> = <string> , [ { LOCATION <id> =  <string> } ] ) ;
```

Syntax 3: Defining the structure of a Location

```
CREATE LOCATION ( <id> = <string> ,
[ { RELATION TYPE <relation_type> = 'directional'|'distance'|'topological',
   RELATION NAME <relation_name> = 'above'|'below'|'leftOf'|'opposite'|
    'rightOf'|'closeTo'|'farFrom'|'contains'|'covers'|'crosses'|
    'disjoint'|'equals'|'overlaps'|'touches',
    OTHER LOCATION <id> = <string> } ] ) ;
```

In addition, one can rename, drop, or alter component definitions. We give an example for each of these queries in the following:

Syntax 4: Renaming a component

```
RENAME COMPONENT <id> = <string>, <new_id> = <string> ;
```

Syntax 5: Dropping a component

```
DROP COMPONENT <id> = <string> ;
```

Syntax 6: Altering the Location component (add a description field)

```
ALTER LOCATION ( <id> = <string> ,
ADD [ DESCRIPTION <description> = <string> ,] ) ;
```

Once the components' definitions are established, we can start creating instances using INSERT queries. Queries 1, 2, and 3 instantiate an infrastructure, a location map, and location components respectively. We do not cover all locations found in Figure 1 to avoid redundancies.

Query 1: Inserting an Infrastructure instance

```
INSERT INFRASTRUCTURE HAVING ( <id> = 'Mall Infra',
LOCATION MAP <id>  = 'Mall Map' ) ;
```

Query 2: Inserting an Location Map instance

```
INSERT LOCATION MAP HAVING ( <id> = 'Mall Map' ,
LOCATION <id> = 'Shop 1', 'Movie Theatre' ) ;
```

Query 3: Inserting two Location instances

```
INSERT LOCATION HAVING ( <id> = 'Shop 1' ,
RELATION TYPE <relation_type> = 'directional',
RELATION NAME <relation_name> = 'leftOf',
OTHER LOCATION <id> 'Movie Theatre' ) ;

INSERT LOCATION HAVING ( <id> = 'Movie Theatre' ,
RELATION TYPE <relation_type> = 'directional',
RELATION NAME <relation_name> = 'rightOf',
OTHER LOCATION <id> 'Shop 1' ) ;
```

In addition, one can select, update, or delete instances of components. We give an example for each of these queries in the following:

Query 4: Selecting all Locations from the Location Map

```
SELECT LOCATION <id> FROM LOCATION MAP WHERE
LOCATION MAP <id> = 'Mall Map';
```

Query 5: Updating the location relation between Shop 1 and Movie Theatre

```
UPDATE LOCATION CHANGE RELATION NAME <relation_name> = 'leftOf'
INTO RELATION NAME <relation_name> = 'opposite',
WHERE ( LOCATION <id> = 'Shop 1', OTHER LOCATION <id> = 'Movie Theatre');
```

Query 6: Deleting a Location

```
DELETE LOCATION WHERE LOCATION <id> = 'Shop 1';
```

This concludes the definition and manipulation syntax/queries for the environment part. Next, we discuss sensor networks, events, and application domains. Due to space limitations, we focus mainly on the syntax to define the components' structures.

5.2 Sensor Network Modeling

The sensor network hosted in the mall comprises of various static and mobile sensors. They monitor the environment properties and produce observations. Some properties/observations are scalar (e.g., temperature) while others are multimedia (e.g., video surveillance). Therefore, we define here the following components: (i) Scalar Property; (ii) Media Property; (iii) Scalar Value; (iv) Media Value; and (v) Sensor. Syntax 7 details the structure of a scalar property which is mapped to a set of scalar observation values. Similarly, a media property (cf. Syntax 8) is mapped to a set of media values and a specific type (e.g., audio, video, image). Syntax 9 defines any scalar observation value produced by a sensor. Each observation has a timestamp, location, related sensor, a datatype, a value, and a unit. Media observation values are detailed in Syntax 10. Each media value is composed of a data object and a set of metadata/value pairs. Similarly to scalar values, each media value has a timestamp, location, and a related sensor. Finally, a sensor is defined as en entity that has a type (e.g., static, mobile), a current location/coverage area, a set of previous locations/coverage areas/capabilities. Each sensor is capable of sensing specific properties and can be hosted on a particular platform (a device or an infrastructure). Syntax 11 describes the sensor component structure.

Syntax 7: Creating a Scalar Property

```
CREATE SCALAR PROPERTY ( <id> = <string> ,
[ { SCALAR VALUE <id> = <string> } ] ) ;
```

Syntax 8: Creating a Media Property

```
CREATE MEDIA PROPERTY ( <id> = <string> ,
[ MEDIA TYPE <id> = 'audio' | 'image' | 'video'  , ]
[ { MEDIA VALUE <id> = <string> } ] ) ;
```

Syntax 9: Creating a Scalar Value

```
CREATE SCALAR VALUE ( <id> = <string> ,
[ DATATYPE <dt> = 'integer' | 'float' | 'boolean' | 'date' | 'time' |
'date time' | 'character' | 'string' ,  VALUE <val> = <empty> , ]
[ UNIT <id> = <string> , ] [ TIMESTAMP <val> = <empty> , ]
[LOCATION <location_id> = <empty> , ] [ SENSOR <sensor_id> = <empty> ] ) ;
```

Syntax 10: Creating a Media Value

```
CREATE MEDIA VALUE ( <id> = <string> ,
[ DATA OBJECT TYPE <dot> = 'audio segment'|'visual segment' ,
DATA OBJECT <do> = <empty> , ]
[ { METADATA <meta> = 'text annotation descriptor'|'fundamental frequency'|
'harmonic descriptor'|'harmonic spectral centroid'|
'harmonic spectral deviation'|'harmonic spectral spread'|
'harmonic spectral variation'|'log attack time'|'power descriptor'|
'spectral centroid'|'spectrum basis'|'spectrum centroid'|
'spectrum envelop'|'spectrum flatness'|'spectrum projection'|
'spectrum spread'|'temporal centroid'|'waveform'|'camera motion descriptor'|
'motion activity descriptor'|'parametric motion descriptor'|
'trajectory descriptor'|'warping parameters'|'bounding box descriptor'|
'point descriptor'|'media duration descriptor'|'media time point descriptor'|
'color layout descriptor'|'color structure descriptor'|
'contour shape descriptor'|'dominant color descriptor'|
'edge histogram descriptor'|'face recognition descriptor'|
'scalable color descriptor' ,  VALUE <val> = <empty> } ,]
[ TIMESTAMP <val> = <empty> , ] [LOCATION <location_id> = <empty> , ]
[ SENSOR <sensor_id> = <empty> ] ) ;
```

Syntax 11: Creating a Sensor

```
CREATE SENSOR ( <id> = <string> ,
( [ HAVING
[ SENSOR TYPE <sensor_type> = 'static' | 'mobile' , ]
[ CURRENT LOCATION <id> = <string> , ]
[ { PREVIOUS LOCATION <id> = <string> , TIME INTERVAL <ti> = <empty> } , ]
[ CURRENT COVERAGE AREA <id> = <string> ,  ]
[ { PREVIOUS COVERAGE AREA <id> = <string> , TIME INTERVAL <ti> = <empty> } ,  ]
[ { CAPABILITY <id> = <string> , VALUE <val> = <string> } ] ] , )
( [ SENSING { SCALAR PROPERTY <id> = <string> |
              MEDIA PROPERTY <id> = <string> } ] , )
( [ HOSTED ON PLATFORM <id> = <string> ] ) ) ;
```

5.3 Event Modeling

Here we detail the event modeling in EQL-CE. We define the event as a n-dimensional space where each dimension mathematically represents an event describing feature. The latter are provided by the application domain (cf. Figure 5). Moreover, event data is the set of sensor observations that help detect the event (i.e., event data belong to the event's n-dimensional space). Therefore, an event has a set of dimensions and event data. In addition, an event also has a set of sensors that provide the required observations for the detection. Finally, we added a type parameter to the event definition to distinguish elementary or atomic events (i.e., that require one observation from one sensor) from complex events (i.e., that require various observations from one sensor), and composite ones (i.e., that require various observations from different sensors). The following syntax defines event modeling components.

Syntax 12: Creating an Event Structure

```
CREATE EVENT ( <id> = <string> ,
[ EVENT TYPE <event_type> = 'elementary' | 'complex' | 'composite' , ]
[ { SENSOR <sensor_id> = <string> } , ]
[ { DIMENSION <dimension> = <string> } , ]
[ { EVENT DATA <data_object> = SCALAR OBSERVATION <so> |
                               MEDIA OBSERVATION <mo> } ] ) ;
```

The following query defines a particular event instance, denoted ’Overheat in Shop 1’, where the three main dimensions are time, location, and temperature. This event relies on scalar temperature observations that surpass the value 30. Once the event instance is defined, any external event detection mechanism (e.g., eVM cf. Figure 3 can use this definition to detect occurrences of this event.

Query 7: Creating an Event Instance

```
INSERT EVENT HAVING ( <id> = 'Overheat in Shop 1' ,
EVENT TYPE <event_type> = 'elementary',
{ SENSOR <sensor_id> = ANY },
{ DIMENSION <dimension> = 'Time', 'Location', 'Temperature' },
{ EVENT DATA <data_object> = SCALAR OBSERVATION <so> } ),
WHERE ( <so>.<id> = 'Temperature',
        <so>.<location_id> = 'Shop 1', <so>.<val> > 30 ) ) ;
```

To keep up with the environment changes (cf. Criterion 4), one could need to re-write obsolete event definitions. Query re-writing is provided automatically by the query optimizer (cf. Figure 3). However, users can request an update at any time. This is illustrated in the following query where we update the event definition provided in Query 7 by only considering observations from Sensor 1.

Query 8: UPDATING an Event Instance

```
UPDATING EVENT CHANGE (
SENSOR <sensor_id> = 'Sensor 1',
WHERE (EVENT <id> = 'Overheat in Shop 1') ) ;
```

5.4 Application Domain Modeling

As previously mentioned in the conceptual layer, application domains have different components, inter-component relations, and targeted events. Therefore, we provide here a generic definition of an application domain related components and relations (denoted Concept, Relation respectively cf. Syntax 13). We also provide a definition for event describing features (cf. Syntax 14) and their datatypes (cf. Syntax 15) that can be instantiated in different domains.

Syntax 13: Creating a Concept/Relation

```
CREATE CONCEPT ( <id> = <string> , [ { ATTRIBUTE <id> } ] ) ;
ATTRIBUTE <id> = CONCEPT <id> | VARIABLE ( <label> = <string> ,
DATATYPE <datatype> = 'integer'|'float'|'boolean'|'date'|'time'|
'date time'|'character'|'string', VALUE <val> = <empty> ) ;

CREATE RELATION ( <id> = <string> ,
[ { CONCEPT <source_id> = <string> } , ]
[ { CONCEPT <target_id> = <string> } ] ) ;
```

Every event feature has an identifier, a set of granularities (e.g., second, minute, hour for time), a function that converts a granularity to another (e.g., 1 minute = 60 seconds), a boolean field indicating if intervals can be created from this feature's values, and a datatype.

Syntax 14: Creating an Event Feature

```
CREATE EVENT FEATURE ( <id> = <string> ,
[ GRANULARITY SET { VALUE <val> = <string> } , ]
[ GRANULARITY FUNCTION <id> = <string> , ]
[ INTERVAL <boolean> = '0' | '1' , ]
[ EVENT FEATURE DATATYPE <event_feature_datatype_id> = <string> ] ) ;
```

An event feature datatype has an identifier, a primitive datatype , a range of allowed values (i.e., lower bound min, upper bound max), and a function that measures the distance between values having the same event feature datatype. These details help translate event describing features (application domain) into dimensions of the event's n-dimensional space (event modeling) using the mediator (cf. Figure 5).

Syntax 15: Creating a Event Feature Datatype

```
CREATE EVENT FEATURE DATATYPE ( <id> = <string> ,
[ DATATYPE <datatype> = 'integer'|'float'|'boolean'|'date'|'time'|
'date time'|'character'|'string', ]
[ RANGE ( MIN <min> = VALUE <val> , MAX <max> = VALUE <val> ) , ]
[ DISTANCE <function_id> = <string> ] ) ;
```

6 CONCLUSION & FUTURE WORK

Many challenges emerge when proposing a EQL adapted to connected environments. In this paper, we addressed the issues of re-usability, and covering various components/functionality. To do so, we proposed EQL-CE: a three layered event query language for connected environments. We detailed its conceptual, logical, and physical layers. EQL-CE users compose EBNF queries, that can be later parsed into SQL, SPARQL, or other languages (re-usability). Our proposal covers various connected environment components (environments, sensor networks, events, and application domains) and functionality (definition, manipulation, access control, event detection). We also proposed a query optimizer that allows query re-writing and the integration of spatial/temporal distribution functions. As future work, we would like to detail the security/privacy related queries and distribution/query re-writing functions. Also, we are currently developing an online simulator to allow users to run tests on a connected environment (e.g., the smart mall). Finally, we want to address additional challenges such as integrating batch queries and continuously processing data streams.

REFERENCES

[1] Yuvraj Agarwal, Bharathan Balaji, Rajesh Gupta, Jacob Lyles, Michael Wei, and Thomas Weng. 2010. Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*. ACM, 1–6.

[2] Darko Anicic, Paul Fodor, Sebastian Rudolph, and Nenad Stojanovic. 2011. EP-SPARQL: a unified language for event processing and stream reasoning. In *Proceedings of the 20th international conference on World wide web*. ACM, 635–644.

[3] Darko Anicic, Paul Fodor, Sebastian Rudolph, Roland Stühmer, Nenad Stojanovic, and Rudi Studer. 2011. Etalis: Rule-based reasoning in event processing. In *Reasoning in event-based distributed systems*. Springer, 99–124.

[4] Arvind Arasu, Shivnath Babu, and Jennifer Widom. 2006. The CQL continuous query language: semantic foundations and query execution. *The VLDB Journal* 15, 2 (2006), 121–142.

[5] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. 2009. C-SPARQL: SPARQL for continuous querying. In *Te 18th international conference on World wide web-WWW'09*. 1061–1062.

[6] Roger S Barga and Hillary Caituiro-Monge. 2006. Event correlation and pattern detection in CEDR. In *International Conference on Extending Database Technology*. Springer, 919–930.

[7] François Bry and Michael Eckert. 2007. Rule-based composite event queries: the language XChange EQ and its semantics. In *International Conference on Web Reasoning and Rule Systems*. Springer, 16–30.

[8] AH Buckman, Martin Mayfield, and Stephen BM Beck. 2014. What is a smart building? *Smart and Sustainable Built Environment* 3, 2 (2014), 92–109.

[9] Sharma Chakravarthy and Deepak Mishra. 1994. Snoop: An expressive event specification language for active databases. *Data & Knowledge Engineering* 14, 1 (1994), 1–26.

[10] EsperTech. [n. d.]. EsperTech. Chapter 5. EPL reference:Clauses. http://esper.espertech.com/release-5.3.0/esper-reference/html_single/index.html#epl_clauses. Accessed: 2019-02-07.

[11] Daniel Gyllstrom, Eugene Wu, Hee-Jin Chae, Yanlei Diao, Patrick Stahlberg, and Gordon Anderson. 2006. SASE: Complex event processing over streams. *arXiv preprint cs/0612128* (2006).

[12] Armin Haller, Krzysztof Janowicz, Simon JD Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Joshua Lieberman, Raúl García-Castro, Rob Atkinson, and Claus Stadler. 2018. The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web* Preprint (2018), 1–24.

[13] Konstantinos Kemalis and Theodores Tzouramanis. 2008. SQL-IDS: a specification-based approach for SQL-injection detection. In *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2153–2158.

[14] Timilehin Labeodan, Christel De Bakker, Alexander Rosemann, and Wim Zeiler. 2016. On the application of wireless sensors and actuators network in existing buildings for occupancy detection and occupancy-driven lighting control. *Energy and Buildings* 127 (2016), 75–83.

[15] Jay Lee, Behrad Bagheri, and Hung-An Kao. 2015. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters* 3 (2015), 18–23.

[16] Elio Mansour, Richard Chbeir, and Philippe Arnould. 2018. eVM: An Event Virtual Machine Framework. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXIX*. Springer, 130–168.

[17] Matthew Perry, Prateek Jain, and Amit P Sheth. 2011. Sparql-st: Extending sparql to support spatiotemporal queries. In *Geospatial semantics and the semantic web*. Springer, 61–86.

[18] Oumy Seye, Catherine Faron-Zucker, Olivier Corby, and Corentin Follenfant. 2012. Bridging the Gap between RIF and SPARQL: Implementation of a RIF Dialect with a SPARQL Rule Engine. *AImWD 2012* (2012), 19.

[19] Winda Astuti Wahyudi and M Syazilawati. 2007. Intelligent voice-based door access control system using adaptive-network-based fuzzy inference systems (ANFIS) for building security. *Journal of Computer Science* 3, 5 (2007), 274–280.

[20] James Welch, Farzin Guilak, and Steven D Baker. 2004. A wireless ECG smart sensor for broad application in life threatening event detection. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, Vol. 2. IEEE, 3447–3449.

[21] Niklaus Wirth. 1977. What can we do about the unnecessary diversity of notation for syntactic definitions? *Commun. ACM* 20, 11 (1977), 822–823.

[22] Niklaus Wirth, Niklaus Wirth, Niklaus Wirth, Suisse Informaticien, and Niklaus Wirth. 1996. *Compiler construction*. Vol. 1. Citeseer.

[23] Johnny KW Wong, Heng Li, and SW Wang. 2005. Intelligent building research: a review. *Automation in construction* 14, 1 (2005), 143–159.

[24] Liyang Yu, Neng Wang, and Xiaoqiao Meng. 2005. Real-time forest fire detection with wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, Vol. 2. IEEE, 1214–1217.

[25] S Zampolli, I Elmi, F Ahmed, M Passini, GC Cardinali, S Nicoletti, and L Dori. 2004. An electronic nose based on solid state sensor arrays for low-cost indoor air quality monitoring applications. *Sensors and Actuators B: Chemical* 101, 1-2 (2004), 39–46.

HSSN: An Ontology for Hybrid Semantic Sensor Networks

Elio Mansour
Univ Pau & Pays Adour
E2S UPPA, LIUPPA
Mont-de-Marsan, 40000, France
elio.mansour@univ-pau.fr

Richard Chbeir
Univ Pau & Pays Adour
E2S UPPA, LIUPPA
Anglet, 64600, France
richard.chbeir@univ-pau.fr

Philippe Arnould
Univ Pau & Pays Adour
E2S UPPA, LIUPPA
Mont-de-Marsan, 40000, France
philippe.arnould@univ-pau.fr

ABSTRACT

Semantic web techniques (e.g., ontologies) have been recently adopted for sensor network modeling. However, existing works do not fully address these challenges: (i) representing different sensor types (e.g., mobile/static sensors) to enrich the network with different data and ensure better coverage; (ii) representing a variety of platforms (e.g., environments, devices) for sensor deployment, thus, integrating new components (e.g., mobile phones); (iii) representing the diverse data (scalar/multimedia) needed for various applications (e.g., event detection); and (iv) proposing a generic model to allow re-usability in various application domains. In this paper, we propose HSSN, an ontology that extends the Semantic Sensor Network (SSN) ontology which is already re-usable and considers various platforms. We extend the representation of sensors, sensed data, and deployment environments to cope with these challenges. We evaluate the consistency, accuracy, clarity, and performance of HSSN.

CCS CONCEPTS

• **General and reference → General conference proceedings**; • **Information systems → Ontologies**; • **Computer systems organization → Sensor networks**.

KEYWORDS

Semantic Sensor Networks, Ontology, Sensor Mobility

ACM Reference Format:
Elio Mansour, Richard Chbeir, and Philippe Arnould. 2019. HSSN: An Ontology for Hybrid Semantic Sensor Networks. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331076.3331102

1 INTRODUCTION

Recently, Sensor Networks (SNs) have impacted more and more application domains [14] such as environmental sensing, military, and medical fields. Various sensors (e.g., camera, microphone) are nowadays embedded in smart phones, and capable of sensing useful data for various purposes (e.g., pollution monitoring in a city). Therefore, considering such devices, and other equipment capable of sensing, is very beneficial for knowledge extraction in sensor networks. Nonetheless, SNs may produce heterogeneous data, that have to be collected, processed, and analyzed in order to provide various services for network managers. Representing, sharing, and integrating the aforementioned data is a challenging task. In order to address this challenge, semantic web techniques, such as ontologies, have been adopted for their information representation. However, existing approaches on sensor network representation [1–4, 6, 11, 13] are restrictive due to the following issues:

- *Lack of platform diversity:* existing approaches do not consider equipment with embedded sensors (e.g., smart phones, drones, machines) as platforms, in addition to traditional platforms (e.g., buildings, cities, offices) where sensors are deployed. Extending the platform representation, by both considering and detailing the representation of various types of platforms, allows the addition of new components to the network, nested platforms, and dynamic, collaborative sensing activities (e.g., crowd-sensing).
- *Lack of sensor diversity:* these works do not represent different sensor types (e.g., mobile/static sensors, simple sensor nodes/multi-sensor devices, sensors capable of sensing scalar/multimedia properties). Providing a more detailed sensor representation that considers various attributes (e.g., mobility) improves network coverage, and allows sensor tracking and dynamic sensing.
- *Lack of data diversity:* most works cover scalar environment properties (i.e., mainly focus on scalar data such as temperature, motion, and neglecting multimedia data such as sounds, images, and videos). Since several devices are capable of sensing both types, and data diversity is required for different application purposes (e.g., event detection), it is important to cover scalar and multimedia data in the representation.
- *Lack of re-usability:* these approaches are heavily linked to a specific application domain. The sensor network modeling should remain generic and re-usable in different contexts.

To answer these challenges, we present here an extension of the widely used Semantic Sensor Network ontology (SOSA/SSN) [7] called HSSN. It allows the representation of hybrid sensor networks, i.e., networks containing mobile/static sensors, scalar/multimedia properties, and infrastructures/devices as platforms where sensors are deployed. We chose to extend SSN since it is already re-usable in various contexts and allows the representation of different platforms. Nonetheless, sensor and data diversity are not fully developed. Our proposal adds diverse data, sensors, and details the description of various platform types. In addition, HSSN does not contain domain specific knowledge and can be easily aligned with other ontologies (e.g., mobile phone, smart building ontologies

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331102

[16]).
The rest of the paper is organized as follows. Section 2 illustrates a scenario that motivates our proposal. Section 3 reviews related work regarding mobility, platforms, and sensed data. Section 4 details the HSSN ontology. Section 5 describes the experimental setup and results. Finally, section 6 concludes the paper and discusses future research directions.

2 MOTIVATING SCENARIO

To highlight the utility of our proposal, we choose the following scenario (we only use this example to concretely illustrate the needs, challenges, and motivations behind our work. We do not consider it to be a generic, all summarizing, sensor network application scenario). Consider a smart mall/shopping center (cf. Fig.1). In order to optimize client comfort, health, and security, the smart mall relies on a set of sensors (s_1-s_9) to monitor the environment. Video surveillance cameras (s_1-s_6) monitor security related events. Humidity, CO_2, and temperature sensors (s_7, s_8, and s_9 respectively) make observations that help regulate the indoor air quality, and temperature. The sensed data is stored and used for these applications. However, many improvements still need to be integrated:

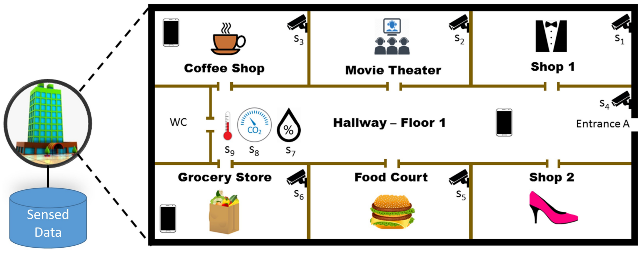


Figure 1: Smart Mall Example

- Need 1- Provide better temperature/air quality readings: relying on measures from a multitude of sensors (instead of only one) allows a more precise monitoring of the environment. Currently, this is not possible since there is only one temperature/air quality sensor in the mall.
- Need 2- Keep track of client positions in the mall since it is useful to know: the number of occupants in each zone, client positions for tracking suspicious/interesting behaviours. Cameras (s_1-s_6) are used by mall agents to monitor limited events and cannot track client locations everywhere.
- Need 3- Cover all areas of the mall: this is critical for client security and safety. In the current setup, many uncovered areas exist (e.g., no temperature monitoring in the movie theater, no video surveillance in Shop 2).
- Need 4- Provide a rich documentation of critical events: in order to increase the understanding of events (e.g., when reporting incidents, providing evidences), rich descriptions should be provided to police with a variety of sensed multimedia and scalar data (e.g., video, audio, image, temperature, humidity). Currently, reports on attack incidents (e.g., gunshot) rely only on video surveillance footage (e.g., no noise levels to confirm the gunshot, no motion data to describe how people ran away). A bigger data variety is needed.
- Need 5- Adapt to changing event detection needs: sometimes new/spontaneous events need to be detected, the mall should be able to sense the required data and detect these events. However, the current sensor configuration/deployment and sensed data cannot be easily modified. This doesn't allow the detection of new events.

In order to address these issues, the mall managers would need to add more sensors to cover all zones. This ensures full coverage of the mall (Need 3), and allows multiple observations from each zone for aggregation (Need 1). In addition, they could replace the cameras with more advanced ones that enable image processing for tracking purposes (Need 2). However, this increases the equipment, maintenance, and implementation costs without addressing Needs 4 and 5. A more appropriate solution would be to integrate visitors' mobile phones (since they embed sensors) as mobile sensors in the mall's network, while avoiding excessive resource consumption from the devices (e.g., draining a phone's battery). This provides the following benefits: (i) sensor mobility provides observations from different areas of the mall, multiple sensors can therefore collaborate to calculate more reliable air quality/temperature measures (Need 1); (ii) mall visitors can easily be tracked using their connected mobile phones (Need 2), location information can also be used to discover uncovered areas (Need 3); (iii) using various sensors from different devices helps cover a wider array of scalar/-multimedia properties (Need 4); and (iv) these devices provide a diversity of hardware (e.g., sensors), software, and services that can be adapted to changing event detection needs (Need 5). However, when adding mobility, diverse data, and devices to the network, the following challenges emerge:

- Challenge 1: How to expressively describe locations in the mall?
- Challenge 2: How to consider ad-hoc devices in the network? How to query them based on their capabilities (e.g., without draining their batteries)? How to represent the services that they provide?
- Challenge 3: How to track locations and coverage areas of mobile sensors?
- Challenge 4: How to collect scalar/multimedia observations from sensors?

Other challenges also exist when modeling sensor networks. However, we address here the aforementioned four challenges from a data modeling perspective by proposing an extension of the semantic sensor network ontology that includes mobility, platform, and data related concepts.

3 RELATED WORK

In this section, we study existing sensor network ontologies. We focus our review on sensor mobility, deployment platforms, and semantic representation of multimedia data. We compare these works based on the following criteria:

(1) *Sensor diversity:* Indicating if different types of sensors exist in the sensor network (e.g., mobile/static sensors, simple nodes/multi-sensor equipment, sensors capable of sensing scalar/multimedia properties).

(2) *Platform diversity:* Stating if the approach allows and details the description of different platforms where sensors are deployed (e.g., in infrastructures, on devices).
(3) *Data diversity:* Denoting the approach's ability to handle various data/properties (e.g., scalar, multimedia).
(4) *Re-usability:* Indicating if the approach is re-usable in various contexts.

3.1 Sensor Diversity

In [2], the authors focus mainly on features that describe the sensor nodes, their functionality, and their current CPU, memory, and power supply states (in order to determine the future state of the WSN). However, they do not represent different types of sensors. In [6], the authors provide a set of ontologies describing missions, tasks, sensors, and deployment platforms for sensor to task assignment. Unfortunately, different types of sensors were not considered. In [7], the authors propose the SOSA/SSN[1] ontologies. Together, they describe systems of sensors and actuators, observations, the used procedures, properties, and so forth. SOSA/SSN propose simple sensor node representation, as well as (sensing) systems/devices. However, SOSA/SSN do not propose any mobility-related concepts, nor multimedia data/properties. The authors only consider one aspect of sensor diversity (i.e., simple sensor nodes/sensor systems). In [1], the authors propose an extension of SSN, denoted MSSN (Multimedia SSN), where they detail the technical aspects of multimedia data (e.g., video, audio segments, frequencies). In this work, the authors improve the sensor diversity of SOSA/SSN by adding a media sensor (i.e., a sensor type that observes multimedia properties). However, they do not achieve full sensor diversity as they do not consider sensor mobility (i.e., mobile/static sensors).

3.2 Platform Diversity

The authors in [9] only consider embedded sensors on mobile phones to monitor noise pollution. In [4], the authors rely on traditional deployment of sensor nodes in the wilderness to detect fire events. The problem is, these works do not provide any platform diversity. In the SSN ontology [7], sensors are deployed on platforms. SSN also introduces systems, that can integrate various sensors, actuators, and samplers. Therefore, SSN provides a foundation for sensor deployment on various platforms (e.g., traditional deployment on platforms, embedding sensors in systems and devices). However, the differences between theses platforms is not detailed in SSN. The description of physical infrastructures/environments such as smart buildings and cities (where it would be interesting to model maps and locations) is different than of machines, drones, and devices that host sensors (where it would be interesting to model hardware and software). It is better to distinguish and detail the description of different platform types to better understand the environments where sensors are deployed (e.g., for location-based services in infrastructures, task assignment based on hardware/-software capabilities for devices). MSSN [1] suffers from the same limitation since it is based on the SSN ontology and does not add any new concepts related to platforms.

3.3 Data Diversity

In [5], the authors represent images for object recognition purposes. The scope of their work does not extend to other types of multimedia data (e.g., video, audio). In [10], the authors are also limited to image representation, since they propose an approach for object-based image retrieval. In [11], the authors monitor noise pollution in urban zones by sensing (audio) noise levels using occupants' mobile phones. The authors only consider noise data, and geo-locations in order to generate a noise level map. Therefore, their proposal does not fully consider data diversity (e.g., video, images, other scalar data). The SSN ontology [7] does not consider multimedia observations. It details scalar sensed data. This motivated the proposal of MSSN [1] where the authors represent multimedia data in sensor networks. For each multimedia observation value, the authors associate data descriptors (denoted media descriptors), and data segments (denoted media segments). Their proposed ontology, MSSN, complements the SSN ontology [7] since the latter does not cover multimedia contents nor multimedia sensors.

3.4 Re-usability

In [9], the authors propose a noise pollution monitoring solution in a city using mobile phones to sense noise. The authors enrich the sensed information by allowing users to add contextual information to their sensor observations. However, it lacks the genericity needed for it to be reusable in other contexts. In [1], the authors propose a multimedia wireless sensor network ontology for event detection purposes (the authors include concepts related to atomic, complex events, and event detection/composition). These added concepts are domain specific and not necessary in other application scenarios. This restricts MSSN's re-usability. Each of these works are task-centric and heavily linked to an application purpose. The SSN ontology [7] remains generic and re-usable in various contexts since it is extensible and does not contain any concepts that link it to any specific application.

3.5 Discussion

The aforementioned works do not fully integrate sensor diversity in their representation of sensor networks (i.e., static/mobile sensors, simple node/multi-sensor devices, and scalar/multimedia sensors). The SSN ontology [7] is a culmination of much of the related work on semantic sensor networks and is the most widely used (reusable). In addition, SSN is extensible, facilitates alignments with other standards, and allows the integration of new concepts. The MSSN ontology [1], integrates multimedia data in SSN. Therefore, we propose to extend SSN since: (i) it partially allows sensor diversity; (ii) it is re-usable and does not contain any domain specific knowledge; and (iii) it allows having various platform types. Moreover, we do not neglect MSSN for its ability to cover multimedia data (data diversity). Therefore, our proposal will extend SSN and use key MSSN concepts in order to achieve full sensor diversity, platform diversity enriched with detailed descriptions of each type (e.g., infrastructures, devices), and finally data diversity through the coverage of scalar/multimedia sensed data.

[1] https://www.w3.org/TR/vocab-ssn/

4 HSSN ONTOLOGY

In this section, we detail our proposed extension of the SSN ontology, and mainly our additions related to: (i) sensor diversity; (ii) platform diversity; and (iii) data diversity. The following prefixes *sosa:*, *ssn:*, *mssn:*, *time:*, and *hssn:* refer to the SOSA[7], SSN[7], MSSN[1], Time[8], and HSSN ontologies respectively. We begin first by describing sensor-related concepts.

4.1 Sensor Diversity

4.1.1 Sensor Mobility. Fig.2 illustrates the sensor types added in HSSN. The concept *Sensor* already exists in the SSN ontology, where mobility is not extensively developed. Therefore, we add two child concepts of *Sensor*: (i) *MobileSensor*, describing any sensor that has the ability to move or change location; and (ii) *StaticSensor*, a sensor that does not change location in time. This allows the sensor network to have diverse sensor types (cf. Criterion 1 - Section 3).

4.1.2 Sensor Tracking. Every sensor has a *Location*. To consider mobility, one should be able to locate any sensor at all times. The object property *isCurrentlyLocatedAt* maps each sensor to its current *Location* (cf. Challenge 3 in Section 2). This is specifically important for tracking mobile sensors, since static sensors do not change locations (cf. Fig.3). A *hasPastLocation* property is added to retrieve the previous positions of a (mobile) *Sensor*, and also a *hasLocationTime* (cf. Fig.4) property is added to map these positions to time instants or intervals in order to track sensors (temporal entities are extracted from Time ontology [8]).

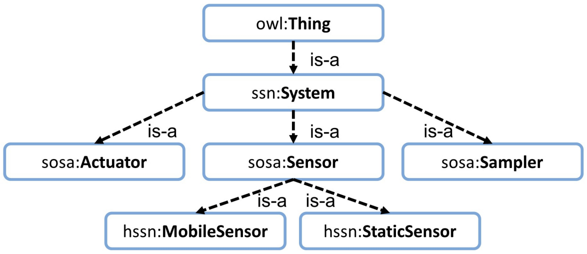


Figure 2: HSSN Sensor View

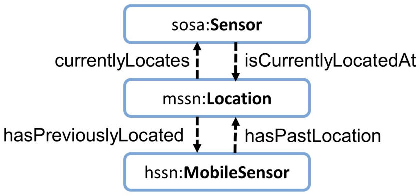


Figure 3: Sensor/Location Mapping

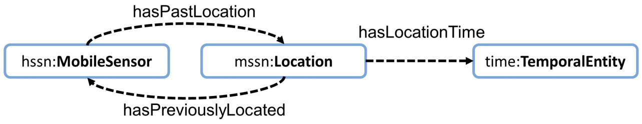


Figure 4: Previous Location/Time Mapping

4.1.3 Coverage Area. Each *Sensor*, mobile or static, has a *CoverageArea* (cf. Fig.5), a geographical zone that contains any sensing activity (i.e., any happening outside of this zone is not detected by the *Sensor*). In order to represent coverage areas, we consider the following: (i) a *CoverageArea* is bound to the sensor's current *Location*; and (ii) the geographical spread of a *CoverageArea* is affected by the sensing range and sensing angles (horizontal and vertical orientation) of the concerned *Sensor*. We represent the coverage area as a sector of space (Fig.6 shows a horizontal slice of the space) where S is the focal point (the sensor's current *Location*), $\alpha, \beta \in [0; 2\pi]$ are the angles that define the horizontal/vertical rotational spread of the coverage area respectively, and the distance $SA = SB$ is the sensing range that defines the extent of the coverage area. The angles and range depend of the sensor's capability properties. For instance, a temperature sensor has $\alpha = \beta = 2\pi$, but a surveillance camera has $\alpha = \frac{\pi}{4}, \beta = \frac{\pi}{6}$ if the camera lens is limited to a 45° horizontal angle, and a 30° vertical angle. Similarly, the sensing range varies from one sensor to another (e.g., 10, 20, 50 meters).

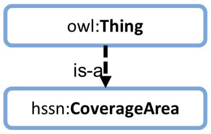


Figure 5: Coverage Area

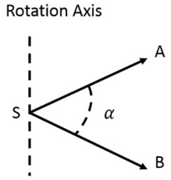


Figure 6: Coverage Area - Horizontal Spread

The composition of a *CoverageArea* is explained in Fig.7. The *SensingLocation* is equivalent to the sensor's *Location*, and the angles and range of the *CoverageArea* are equivalent to the sensor's *HorizontalAngle*, *VerticalAngle*, and *Range* properties that we added in HSSN as part of a system's properties. Since static sensors are immobile, it is easy to know their coverage areas using the sensor's location, and its sensing range and angles. In contrast, knowing the coverage areas of mobile sensors is more challenging, since these areas move when the sensors move. In order to keep track of these changes, the object property *currentlyCovers* maps each *Sensor* to its current *CoverageArea* (cf. Fig.8). Also, the property *hasPastCoverageArea* maps mobile sensors to their respective sets of previous coverage areas (cf. Challenge 3 in Section 2). Finally, *hasCoverageTime* is the property that maps previous coverage areas to temporal entities (i.e., time instant or interval from Time ontology [8]) for tracking purposes (cf. Fig.9).

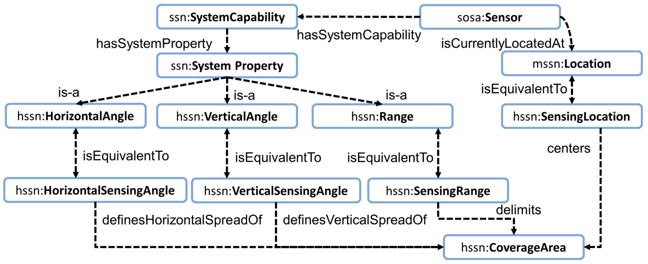


Figure 7: Coverage Area Composition

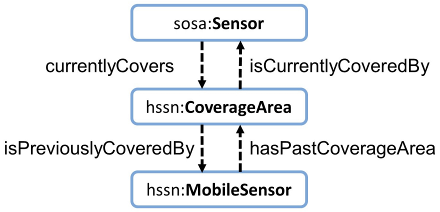


Figure 8: Sensor/Coverage Area Mapping

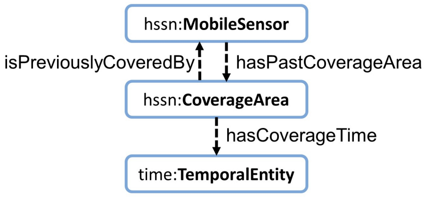


Figure 9: Coverage Area/Time Mapping

4.2 Platform Diversity

4.2.1 Infrastructure Representation. In SSN[7], sensors are deployed on platforms. In Fig.10, we define the following child concepts of *Platform*: (i) *Infrastructure*, a physical environment having locations where sensors could be deployed (cf. Challenge 1 in Section 2); and (ii) *Device*, an electronic equipment where sensors could be embedded (cf. Challenge 2 in Section 2). This allows different types of deployments such as the traditional deployment in environments (e.g., buildings, malls) or nested deployment of multi-purpose devices that in turn embed sensors (e.g., mobile phones). This provides platform diversity (criterion 2 cf. Section 3). Every *Infrastructure* describes a specific physical environment where sensors are deployed. Therefore, infrastructures can host platforms such as other infrastructures (e.g., cities host buildings) and devices (e.g., buildings host mobile phones). However, devices can embed systems of sensors, actuators, and samplers but cannot host infrastructures (e.g., buildings). Each *Infrastructure* is described by a *Location Map* which contains (*isComposedOf* property) a set of *Locations* (cf. Fig.11). For example, a building is an *Infrastructure* that has a *LocationMap*. The latter describes the spatial relations between individual *Locations* in the building such as floors, offices, etc. HSSN uses topological, distance, and directional relations to describe the spatial ties that exist between individual *Locations*. We integrate the aforementioned location-related concepts in order to locate sensors, and better understand the spatial constraints/setup of the *Infrastructure*.

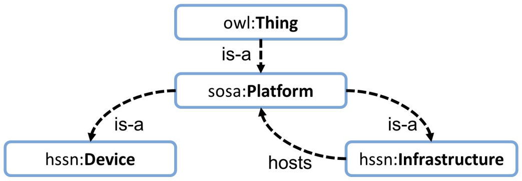


Figure 10: Platform Representation

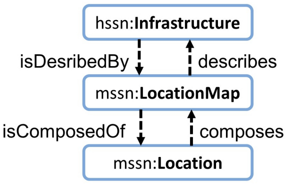


Figure 11: Infrastructures

4.2.2 Device Representation. A *Device* is another type of *Platform* where sensors are deployed. It is introduced in HSSN to represent mobile phones and other sensing equipment. A *Device* has sub-concepts for storage, communication, processing, and power supply, in addition to the ability of embedding sensors (using the *deployEntity* concept cf. Fig.12). These concepts describe the *Hardware* of a *Device*. The *Software* part is also represented. A *Device* could be used for various purposes (e.g., representing mobile phones for mobile phone sensing, machines with mounted sensors for fault detection in an Industry 4.0 scenario). The hardware and software representation allows complex queries such as assigning sensing tasks to devices based on their processing capabilities, or battery status (cf. Challenge 2 in Section 2). Finally, each *Device* can provide a set of services. Fig.13 illustrates our service modeling, inspired by the Web Service Modeling Ontology (WSMO) [12]. We created generic concepts that can be aligned with WSMO. We do not aim to detail the service description to allow alignments with any other service ontology. We limit the service modeling to the following concepts: Service *Metadata* describes the properties of a *Service*. The *Input* represents the set of variables and constraints required for correct service execution, while the *Output* is the set of generated results. The functionality of a service is described by the *Capability* concept which is mapped to a specific *UserGoal* or objective (i.e., a user desire satisfied by the service). Users communicate with a service through *UserInteractionInterfaces* (choreography in WSMO). Finally, services communicate with each other via the *ServiceInteractionInterface* (service orchestration in WSMO). Finally, the infrastructure and device detailing also improves sensor diversity by allowing the representation of simple sensor nodes in infrastructures, multi-sensor systems, and multi-sensor devices.

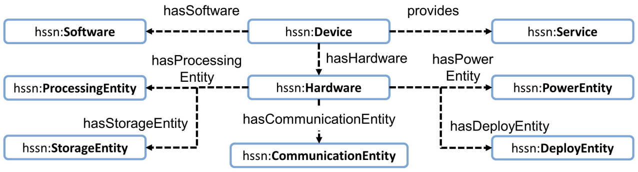


Figure 12: Device Components

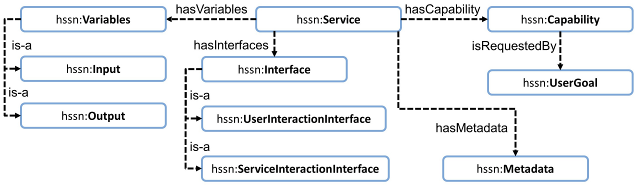


Figure 13: Service Components

4.3 Data Diversity

Audio, image, and video data can be sensed by mobile or static sensors (e.g., surveillance cameras, mobile phones). Also, in order to detect complex events (e.g., gunshot) a combination of multimedia and scalar observations is needed. Therefore, we aim to integrate concepts related to multimedia properties (cf. Criterion 3 in Section 3). In MSSN [1], multimedia data/properties are integrated in SSN. We re-organize MSSN multimedia concepts into scalar (e.g., temperature, motion) and multimedia (e.g., noise, video) properties as illustrated in Fig.14. Also, we introduce in Fig.15 the *mediaSenses* and *scalarSenses* relationships to map sensors to their corresponding scalar and/or multimedia observable properties (cf. Challenge 4 in Section 2). This highlights the sensor diversity in HSSN since static/mobile sensors can detect scalar and/or multimedia properties. The authors in [1] also describe technical aspects/metadata of multimedia objects such as annotations, audio (e.g., frequencies), motion (e.g., trajectories), visual (e.g., color histograms). We use these concepts in HSSN to describe sensor observation values. A *MediaValue* in HSSN is composed of the *MultimediaData* concept, referring to the audio, video, or image objects/files and the *MediaDescriptor* concepts, describing the metadata of the multimedia objects (e.g., frequencies, colors). *ScalarValues* are textual (e.g., temperatures, humidity levels). Finally, we map observation values to their related properties using the *hasMediaValue* and *hasScalarValue* relationships. Sensors can now be correctly mapped to observable properties and observation values (cf. Challenge 4 in Section 2).

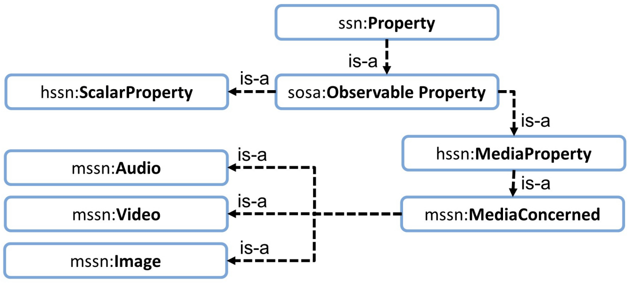


Figure 14: Observable Properties

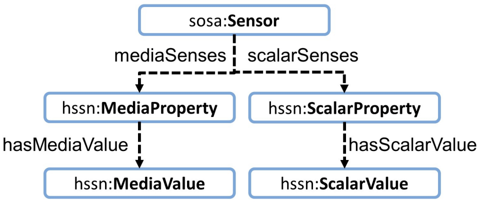


Figure 15: Sensors/Properties

In conclusion, new concepts and properties are introduced in HSSN in order to address the challenges presented in Section 2. Our proposal details the representation of infrastructures (a type of platforms) by adding location maps, individual locations, and spatial relations. This allows the expressively describe locations (cf. Challenge 1). In HSSN we describe devices as platforms that host sensors. We detail device hardware, software, and provided services. In addition, we add properties that help locate, track, and query these devices (cf. Challenge 2). HSSN also provides a description of sensor coverage areas and properties that map both locations and coverage areas to mobile/static sensors at any time (cf. Challenge 3). Finally, we address data heterogeneity by detailing multimedia data objects, their metadata, and scalar data. We also map them to their respective sensors (cf. Challenge 4).

5 IMPLEMENTATION AND EXPERIMENTAL SETUP

5.1 HSSN Implementation

We implemented the HSSN ontology using Protege 5.2.0[2]. The files are available at http://spider.sigappfr.org/research-projects/hybrid-ssn-ontology/ (External Links - Download ontology files). Also, a complete documentation can be found at http://spider.sigappfr.org/HSSNdoc/index-en.html. In the following, we detail the SPARQL queries used during the experimentation. Then, we describe the experimental setup, before discussing the obtained results from an accuracy, clarity, performance, and consistency standpoint.

5.2 Illustration Example

The challenges mentioned in Section 2 can be addressed via the following SPARQL queries: **Platform Diversity:** In order to expressively describe locations (Challenge 1) in the mall infrastructure, a detailed representation of location maps and locations is needed (Query 1). Also, covered and uncovered areas should be easily found (Query 2). In order to consider ad-hoc devices in the network (Challenge 2), one should be able to query devices, their hardware (e.g., embedded sensors), software, and services. Query 3 shows how to locate a mobile device by querying its embedded sensor. Similarly, one could query a device based on other characteristics (e.g., battery status, processing power).

Query 1: Knowing the spatial description of infrastructures

```
SELECT distinct ?infrastructure ?locationmap ?location WHERE
{?infrastructure isDescribedBy ?locationmap. ?locationmap isComposedOf ?location.}
```

[2] https://protege.stanford.edu/

Query 2: Knowing covered locations

```
SELECT distinct ?location ?coveragearea WHERE {?location isIncludedIn ?coveragearea.}
```

Query 3: Locating mobile devices, querying device hardware

```
SELECT distinct ?location ?dev WHERE {?location currentlyLocates ?sensor.
?sensor isEmbeddedOn ?du. ?du hasExpansionCard ?hd. ?hd isRelatedToDevice ?dev.}
```

Sensor Diversity: To track sensors at all times (Challenge 3), it is important to know current locations/coverage areas for all sensors (Query 4), as well as previous ones (Query 5).

Query 4: Finding current sensor locations/coverage areas

```
SELECT distinct ?location ?sensor ?coveragearea WHERE
{?location currentlyLocates ?sensor. ?sensor currentlyCovers ?coveragearea.}
```

Query 5: Finding previous sensor locations

```
SELECT distinct ?location ?sensor WHERE {?location hasPreviouslyLocated ?sensor}
```

Data Diversity: In order to consider data diversity (Challenge 4), on should be able to distinguish scalar/multimedia data and correctly map them to sensors (Queries 6 and 7).

Query 6: Mapping sensors to their scalar properties and observations

```
SELECT distinct ?sensor ?property ?observation WHERE
{?sensor scalarSenses ?property. ?property isScalarValueOf ?observation.}
```

Query 7: Mapping sensors to their multimedia properties and observations

```
SELECT distinct ?sensor ?property ?observation WHERE
{?sensor mediaSenses ?property. ?property isMediaValueOf ?observation.}
```

5.3 HSSN Experimental Setup

Here, we did not aim to experiment SSN concepts and properties. We evaluated the impact of our newly added concepts (e.g., static/mobile sensors, infrastructures/devices, multimedia/scalar data). Our objectives were the following:

(1) *Accuracy Evaluation:* Checks if the added concepts/properties answer the aforementioned challenges. This query based evaluation highlights the impact of our extensions in overcoming the challenges mentioned in Section 2.

(2) *Clarity Evaluation:* Checks if the labels used to describe the concepts/properties are clear and unambiguous to domain stakeholders. The aim is to evaluate the compatibility and clarity of our provided description with respect to the application domain.

(3) *Performance Evaluation:* Measures the impact of HSSN additions on performance (i.e., query run time). The aim is to evaluate the feasibility, performance-wise, of integrating HSSN in sensor network applications.

(4) *Consistency Evaluation:* Checks if the added concepts/properties generate inconsistencies (e.g., anti-patterns) within the structure of the ontology. The aim is to evaluate the soundness of the ontology graph.

5.3.1 *Accuracy Evaluation.*

We created a population of individuals and ran the aforementioned queries. Then, we compared the obtained and expected results. We created two infrastructures, each described by a location map containing 500 locations. Then, 1000 sensors were deployed (500 mobile/static, 500 scalar/media). Each sensor is located in one location, covers one coverage area, observes one property, and produces one observation value.

Platform Results: We ran queries 1, 2, and 3. The returned results match perfectly the expected ones. Infrastructures were correctly assigned to their location maps and included locations. This allowed the identification of distinct spaces/areas. Query 2 correctly returned the set of distinct locations included in each coverage area. This allowed the identification of non covered locations. Query 3 allowed the identification of device hardware related to the embedded sensors. Also, the mobile devices were correctly located in the location map.

Mobility Results: We ran queries 4 and 5 on the population of individuals and for each case the returned results matched exactly the expected ones. Sensors were correctly assigned to their current/previous locations and coverage areas.

Data Results: We ran queries 6 and 7 and obtained an exact matching between the actual and expected results. Thus, scalar/multimedia properties were correctly distinguished. Also, sensors were correctly assigned to the scalar or multimedia observations that they produced.

Result Discussion: The test results showed that locating any type of sensor (i.e., simple node/multi-sensor device, static/mobile sensors, and scalar/multimedia sensors), and knowing their coverage areas is possible at any point in time. Hence, allowing tasks such as tracking mobile sensors, and detecting uncovered areas. Also, the results showed that the detailing of infrastructure and device descriptions (platform diversity) allowed a better knowledge of the environment space (also important for locating sensors). Multi-sensor devices were also detailed by describing their hardware and software which proved useful when querying devices based on their capabilities (e.g., we ran an additional query that returns sensors/devices with good battery status). From a data diversity standpoint, the results showed that sensors that sense multimedia/scalar properties were correctly distinguished and their observations were accurately retrieved. To conclude, the query results confirmed that the added extensions (i.e., regarding sensor, platform, and data diversity) accurately answer the challenges mentioned in Section 2.

5.3.2 *Clarity Evaluation.*

We created two evaluation forms: the first[3] for evaluating the ambiguity of the labels used to describe the HSSN concepts, and the second[4] for evaluating the ambiguity of the labels used to describe inter-concept relations. We sent

[3] Link: https://goo.gl/forms/blc8pKLLqtNtjXHI2

[4] Link: https://goo.gl/forms/KNNY3XsmGp0ptM2N2

the two forms to 50 sensor network and ontology experts (25 networking experts, and 25 computer scientists). Results in Fig.16 and 17 show that terms considered clear by computer scientists are sometimes found ambiguous by network experts and vice-versa. Fig.16 shows that a few terms do not meet the acceptable ambiguity level (e.g., ComUnit, DeployUnit), while others (e.g., MediaProperty, MediaValue) need some clarification. Therefore, we considered the experts' suggestions in the final version of the ontology by modifying the following: (i) *ExpansionCard* instead of *DeployUnit*; (ii) *PowerSupply* instead of *PowerUnit*; (iii) *NetworkInterface* instead of *ComUnit*; (iv) *Memory* instead of *StorageUnit*; (v) *Processor* instead of *ProcessingUnit* ; and (vi) *Multimedia* instead of *Media.* Finally, Fig.17 shows that in most cases, both categories of experts assigned correctly the inter-concept relationships. Networking experts have low success on the first two questions since the latter are outside of their domain of expertise (regarding inheritance between concepts).

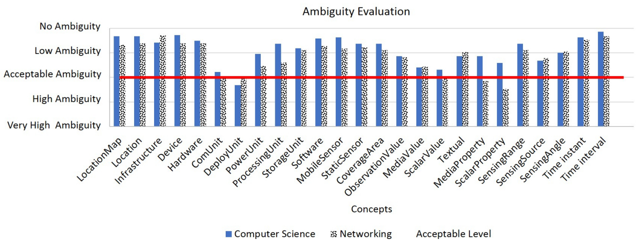


Figure 16: Concept Evaluation

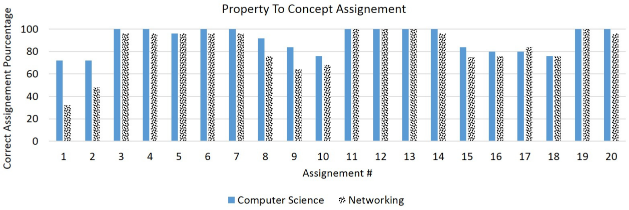


Figure 17: Property Evaluation

Result Discussion: The clarity evaluation allowed the identification and correction of ambiguous/unclear labels that we used to describe our added concepts/properties. In the version currently available online, all labels achieve an acceptable level of clarity (based on the stakeholders' feedback). This reinforces the re-usability of HSSN since it is unambiguous and easily understood.

5.3.3 *Performance Evaluation.*

Performance Evaluation. In order to evaluate the performance of HSSN, we measured the query run-time by running each of the previously mentioned queries 10 times and calculating the average. We varied the size of the population (100 sensors, 1000 sensors, and 10000 sensors) in order to test various scenarios related to mobility, platforms, and data.

Mobility impact: In this test, we varied the percentage of mobile sensors in the network (0, 30, 50, 70, and 100 %). Then, we retrieved the current/previous sensor locations (cf. Fig.18 and 19). We measured the run-time for queries 4 and 5. In Fig.18, we noticed that increasing the number of mobile devices increases the time required to retrieve current sensor locations. This is due to the fact that locating a device (Query 3) was a more complex task than locating a static sensor since we needed to locate the sensor, its deployment unit, hardware, and then the device. We noticed the same pattern for all three cases (100, 1000, 10000 sensors). Finally, the progression from 0% to 100% mobile devices had a quasi-linear impact on query run-time. Similarly, Fig.19 details the query run-time for retrieving previous different sensor locations. Since mobile sensors have a larger list of previous locations in comparison with static sensors, increasing the mobility percentage (0, 50, 100 %) increases the query run-time. This progression was also quasi-linear for all three cases (100, 1000, 10000 sensors).

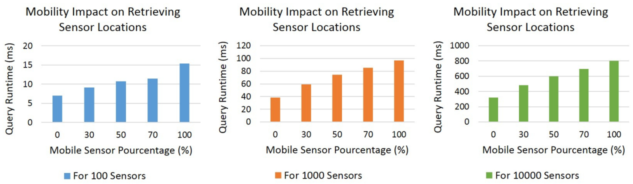


Figure 18: Mobility impact on current location retrieval

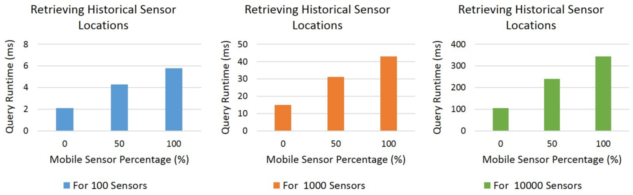


Figure 19: Mobility impact on previous location retrieval

Platform impact: In this test, we varied the sensor distribution on the platform locations. We tested three different scenarios (i) each sensor is located in one location; (ii) all sensors are located in one location; and (iii) half of the sensors are located in a location and the other half in another. We measured the run-time of the query that retrieves sensor locations.

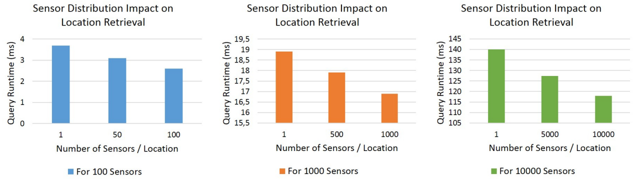


Figure 20: Platform impact on current location retrieval

Fig.20 shows how sensor distribution on locations affected the time needed to map sensors to their current locations. When all sensors were located in one location, the required time to perform this task was minimal. Then, as we began to decrease sensor densities, the query took more time. Finally, the worst case was when every location contained only one sensor.

Data impact: Here, we checked the impact of scalar/multimedia data on the run-time of queries 6 and 7 (cf. Fig.21).

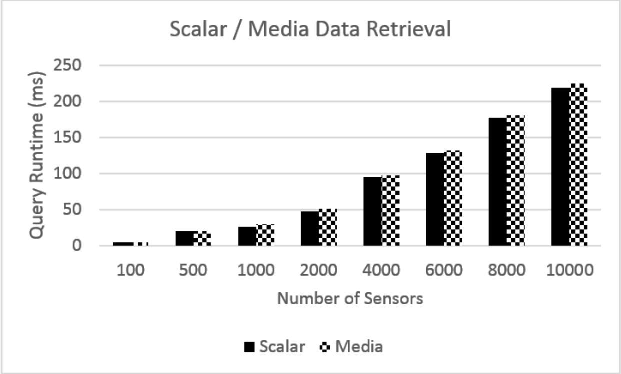


Figure 21: Data impact on observation retrieval

For data diversity impact on performance (cf. Fig.21), we noticed that in all cases (100, 1000, 10000 sensors) the query run-time was similar when considering scalar and multimedia data. This is due to the fact that we were measuring the time required to retrieve the data and not the time needed to capture/sense it.

Result Discussion: The performance evaluation showed that the added concepts/properties do not heavily impact the query run time, which remains quasi-linear in most cases. This highlights the feasibility of using of HSSN in sensor applications (from a performance point of view).

5.3.4 Consistency Evaluation. In [15], consistency is defined as a criterion that verifies if the ontology allows contradictions. The descriptions in the ontology should be consistent.

Consistency Queries: To evaluate consistency, we adopted the following SPARQL queries that search for anti-patterns, a strong indicator of inconsistencies, in the ontology. Query 8 detects concepts with no parent, and query 9 detects abnormally disjointed concepts in the ontology:

Query 8: Searching for concepts with no parent

```
SELECT ?a WHERE {?a subClassOf owl:Nothing.}
```

Query 9: Searching for abnormally dijointed concepts

```
SELECT distinct ?A ?B1 ?B2 ?C1 WHERE
{?B1 subClassOf ?A. ?B2 subClassOf ?A. ?C1 subClassOf ?B1. ?C1 disjointWith ?B2.}
```

Results & Discussion: We found no inconsistencies in the HSSN ontology structure. The only concept subsuming nothing is owl:Nothing (Query 8). Query 9 results indicate that there are no concepts that have abnormal disjoint relations with their relatives. This denotes the soundness of the integration of newly added concepts mainly with the SSN core. Finally, to conclude the inconsistency evaluation, we ran Protege's HermiT 1.3.8.413 reasoner, and found no inconsistencies between the asserted class hierarchy and inferred one. This highlights the soundness of the graph structure, which proves critical when considering future alignments between HSSN and other ontologies (e.g., that describe smart buildings, events).

6 CONCLUSION & FUTURE WORK

Many works adopted ontologies for better semantic representation of sensor networks. These approaches do not fully consider diversity in terms of sensors, data, platforms, and application purposes. In this paper, we propose an extension of the Semantic Sensor Network ontology (SSN), since it is already re-usable in various contexts. Our proposed ontology, denoted HSSN, adds to SSN sensor mobility, and multimedia data related concepts in order to have a representation of hybrid sensor networks. HSSN also extends the platform representation of SSN in order to fully consider platform diversity. We implemented HSSN, evaluated the consistency, accuracy of our additions, and their impact on performance. As future work, we would like to continue the ongoing evaluation of the completeness of the ontology through comparisons with mobility and sensor taxonomies. Finally, we want to represent a sensor network in a smart environment (e.g., smart building, city) for event detection purposes.

REFERENCES

[1] Chinnapong Angsuchotmetee, Richard Chbeir, and Yudith Cardinale. 2018. MSSN-Onto: An ontology-based approach for flexible event processing in Multimedia Sensor Networks. *Future Generation Computer Systems* (2018). https://doi.org/10.1016/j.future.2018.01.044

[2] Sasikanth Avancha, Anupam Joshi, Chintan Patel, et al. 2004. Ontology-driven adaptive sensor networks. *MobiQuitous 2004* (2004), 194–202.

[3] Payam Barnaghi, Stefan Meissner, Mirko Presser, and Klaus Moessner. 2009. Sense and sens'ability: Semantic data modelling for sensor networks. In *Conference Proceedings of ICT Mobile Summit.*

[4] David M Doolin and Nicholas Sitar. 2005. Wireless sensors for wildfire monitoring. In *Smart Structures and Materials 2005: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, Vol. 5765. International Society for Optics and Photonics, 477–485.

[5] Nicolas Durand, Sebastien Derivaux, Germain Forestier, Cedric Wemmert, Pierre Gançarski, Omar Boussaid, and Anne Puissant. 2007. Ontology-based object recognition for remote sensing image interpretation. In *Tools with Artificial Intelligence, 2007. ICTAI 2007.*, Vol. 1. IEEE, 472–479.

[6] Mario Gomez, Alun Preece, Matthew P Johnson, Geeth De Mel, Wamberto Vasconcelos, Christopher Gibson, Amotz Bar-Noy, Konrad Borowiecki, Thomas La Porta, Diego Pizzocaro, et al. 2008. An ontology-centric approach to sensor-mission assignment. In *International Conference on Knowledge Engineering and Knowledge Management.* Springer, 347–363.

[7] Armin Haller, Krzysztof Janowicz, Simon JD Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Joshua Lieberman, Raúl García-Castro, Rob Atkinson, and Claus Stadler. 2018. The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation. *Semantic Web - Interoperability, Usability, Applicability an IOS Press Journal* (2018).

[8] Jerry R Hobbs and Feng Pan. 2006. Time ontology in OWL. *W3C working draft* 27 (2006), 133.

[9] Nicolas Maisonneuve, Matthias Stevens, and Bartek Ochab. 2010. Participatory noise pollution monitoring using mobile phones. *Information Polity* 15, 1, 2 (2010), 51–71.

[10] Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G Strintzis. 2003. An ontology approach to object-based image retrieval. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, Vol. 2. IEEE, II–511.

[11] Rajib Kumar Rana, Chun Tung Chou, Salil S Kanhere, Nirupama Bulusu, and Wen Hu. 2010. Ear-phone: an end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks.* ACM, 105–116.

[12] Dumitru Roman, Uwe Keller, Holger Lausen, Jos De Bruijn, Rubén Lara, Michael Stollberg, Axel Polleres, Cristina Feier, Cristoph Bussler, and Dieter Fensel. 2005. Web service modeling ontology. *Applied ontology* 1, 1 (2005), 77–106.

[13] David J Russomanno, Cartik Kothari, and Omoju Thomas. 2005. Sensor ontologies: from shallow to deep models. In *System Theory, 2005. SSST'05.* IEEE, 107–112.

[14] Kazem Sohraby, Daniel Minoli, and Taieb Znati. 2007. *Wireless sensor networks: technology, protocols, and applications.* John Wiley & Sons.

[15] Steffen Staab and Rudi Studer. 2010. *Handbook on ontologies.* Springer Science & Business Media.

[16] Thanos G Stavropoulos, Dimitris Vrakas, Danai Vlachava, and Nick Bassiliades. 2012. Bonsai: a smart building ontology for ambient intelligence. In *Proceedings of the 2nd international conference on web intelligence, mining and semantics.* ACM, 30.

Enabling Propagation in Web of Trust by Ethereum

Francesco Buccafurri
University Mediterranea of Reggio Calabria
Reggio Calabria, Italy
bucca@unirc.it

Lorenzo Musarella
University Mediterranea of Reggio Calabria
Reggio Calabria, Italy
lorenzo.musarella@unirc.it

Roberto Nardone
University Mediterranea of Reggio Calabria
Reggio Calabria, Italy
roberto.nardone@unirc.it

ABSTRACT

Web of Trust offers a way to bind identities with the corresponding public keys. It relies on a distributed architecture, where each user could play the role of certificate signer. With the widespread diffusion of social networks, the trust propagation is a matter of growing interest. This paper proposes an approach enabling the propagation in Web of Trust by means of Ethereum. The usage of Ethereum eliminates the necessity of single-organization trusted services, which is, in general, not realistic. Although the information stored on Ethereum is public, the privacy of users is protected because trust chains involve only Ethereum addresses and strong measures are implemented to contrast their malicious de-anonymization. The approach relies on the usage of a smart contract for storing the status of certificate signatures and to manage revocations. When a user u wants to trust another user v, the smart contract checks the presence of trust chains originating from root nodes of u.

CCS CONCEPTS

• **Security and privacy** → **Trust frameworks**; *Key management*; *Social network security and privacy*; **Trust frameworks**; *Social network security and privacy*;

KEYWORDS

Trust Propagation, Blockchain, Ethereum, Smart Contract, Social Network, Pretty Good Privacy.

ACM Reference format:
Francesco Buccafurri, Lorenzo Musarella, and Roberto Nardone. 2019. Enabling Propagation in Web of Trust by Ethereum. In *Proceedings of 23rd International Database Engineering Applications Symposium, Athens, Greece, June 10–12, 2019 (IDEAS'19),* 6 pages.
https://doi.org/10.1145/3331076.3331108

1 INTRODUCTION

The widespread diffusion of social and recommendation systems have experienced exponential growth in recent years. These systems offer very attractive means of social interactions and communications, but also threats for security concerns. Confidentiality, for example, is weakened by the lack of key management frameworks that are able to bind social identities with the corresponding public keys. Consequently, the risk of malicious events is very high. For this reason, we believe that more effective solutions and mechanisms are required when users, in open environments, rely on public key encryption to obtain security services. For example, a user should get answers to the following questions: "is the person I am talking to really the one she/he claims to be?", "who ensure the trust level of my recipient?", "is there somebody embodying the recipient?".

The design of a central authority, which is trusted by everyone, is often not applicable in these contexts. On the contrary, *Web of Trust* [13] ensures a higher level of flexibility since it adopts a distributed approach that better suits the nature of the context we are referring to. Indeed, Web of Trust offers a way to bind identities with the corresponding public keys in the form of certificates without relying on central authority and exploiting the direct trust between users. In Web of Trust, users have the capability to sign each other's certificates (i.e., the couple identity - public key), and this mechanism originates a directed trust graph in which arcs represent signatures. When a user needs to obtain information about a certificate issued by an unknown user, he/she has to check for the presence of one or more trusted parties in the list of signatures associated with that certificate.

Although Web of Trust allows in principle trust propagation, its direct implementation into the current architecture would require either the adoption of certificates with size exponentially growing with propagation or trusted servers to which users delegate trust chain verification. With no propagation, no trust is required for servers. Distributed ledgers offer a solution to avoid trusted central authorities and to guarantee the storage of shared information in an immutable and distributed way. In this paper, we propose an approach that enables trust propagation in Web of Trust and exploits Ethereum to work as a public key infrastructure holding the list of signatures and to implements trust management. This approach matches the current state of Pretty Good Privacy (PGP) [23] public key server infrastructure. The result is a system where users can sign certificates of other users and can retrieve information about the trust level associated with a certificate by consulting the blockchain. Moreover, the proposed solution does not require the disclosure of social identities both for the signature and for the verification of trust phases, since it is based on users' pseudonyms given by the corresponding Ethereum addresses.

The paper is structured as follows. Section 2 gives some background about PGP and Web of Trust, motivating our proposal; Section 3 offers details about our solution; Section 4 discusses the implementation issues; Section 5 gives some conclusive remarks and offers hints for future works.

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331108

2 BACKGROUND AND MOTIVATION

2.1 Web of Trust

Nowadays, we are spectators of an incredible growth of online social networks (OSNs). Unfortunately, at the same time, risks and attacks towards these systems are increasing [3, 7]. The risks associated with OSNs are of a different type. In this paper we focus on the trust propagation, that is orthogonal and complementary to some other problems such as the identification of fake accounts to contrast social attacks. In fact, a lot of works can be found in this area ([4, 7, 21] to cite a few). Instead, in this work, we propose an approach that enables trust propagation in the Web of Trust for secure communications by exploiting Ethereum and the smart contract properties.

One of the most known methods that provide a mean to trust the association between identities and public keys is the "Web of Trust". It has been firstly proposed in the context of Pretty Good Privacy (PGP) in 1991 by Phil Zimmermann [23]. PGP offers authentication and privacy protection for data communication and Web of Trust which consists of a decentralized method for certifying a given PGP public-key certificate. Indeed, people can sign each other's public key so that progressively, and dynamically, they create a network of interconnected links and signatures [1]. When someone needs to trust the public key of an unknown user, she/he has to verify the set of signatures and of the signatures' identities associated with it. The result is that each person will have a different and subjective idea of the other one based on signatures of her/his certificate. This is due to the figure of "introducers", who are the trusted people whose signature represents a trustworthiness guarantee for a PGP certificate. The set of introducers is chosen by each person, so influencing the personal perception of the network identities.

In this context, it is fundamental to compute and manage, in a timely manner, the trustworthiness of both PGP public-key certificate and introducers. According to [1], the former can have three levels of trustworthiness (i.e., *undefined*, *marginal* and *complete*), while the latter can be *fully*, *marginal*, *untrustworthy* or *don't know* trusted. Moreover, since Web of Trust is generally subjective, every user is able to resolve her/his scepticism by tuning suitably two thresholds that are related to the minimum number of introducers' signatures that need to present a certificate to be considered as *complete* by a user.

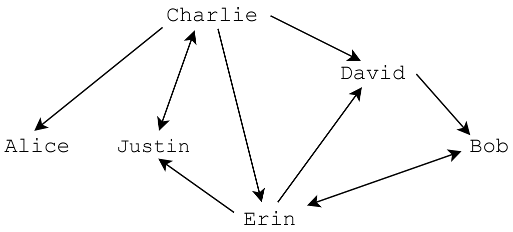


Figure 1: Signatures in Web of Trust

To explain the basic idea behind the Web of Trust, let us consider the scenario depicted in Figure 1. A total of 6 users are interconnected by a set of edges representing signatures of public keys. The origin on an edge represents the signer, while the destination is the signed party. The edge is bidirectional when a mutual signature is present. For example, *Charlie* signs the certificates of *Alice*, *David*, *Erin* and *Justin*, while *Bob* is signed by *David* and *Erin*. The reader can deduce the other signatures by following the edges.

In this scenario the verification of a user's public key can be asked from each user after the definition of her/his own set of introducers. For sake of simplicity, we do not consider the levels of trustworthiness. It means that, for example, *Alice* could trust the *Bob*'s public key if *David* and/or *Erin* belong to the set of *Alice*'s introducers and she has settled for 1 signature. If she needs at least 2 signatures, both *David* and *Erin* needs to belong to her set of introducers; if she needs more than 2 signatures to trust a user's public key, she will never trust *Bob*'s public key.

Furthermore, what happens if *Alice* wants to verify *Bob*'s public key and her introducer is only *Charlie*? In our previous example, there is a path of signatures (of 2 steps) from *Charlie* to *Bob*. PGP is equipped with a parameter, named CERT_DEPTH, to establish the maximum length of the certification chain; unfortunately, this value is often not used in real applications since it can be quite difficult to use in an appropriate way [1, 16]. This led to decrease in the interest of researchers in working on new solutions for the propagation of trust. Anyway, since we are talking about Web of Trust, we think it is necessary to enable propagation to make the most of PGP, especially in OSN applications where the number of users is very high. In this sense, we can start from the assumption on "objectiveness" of trust, such that, if *Alice* fully trusts *Charlie* as her introducer, she is trusting, at the same time, his actions and his decisions. As a consequence, if *Charlie* considers *David* trustworthy, then *Alice* will consider *David* trusted as well.

In a standard PGP scenario, enabling propagation leads to keep an updated and trusted certification chain in each PGP member's certificate, and, moreover, it requires to store full trust chains on PGP servers. This is, clearly, totally in contrast with the Web of Trust principles. For this reason, the rest of the paper will describe our solution that exploits Ethereum instead of PGP servers.

To the best of our knowledge, few works proposed ways to enable propagation in Web of Trust or use a blockchain-based solution to implement PGP and trust evaluation mechanisms. In [15], for example, the authors proposed a way to further expand the trusted neighbourhood. The basic idea relies on the possibility to sign a user's certificate with the couple of values $\{+1, -1\}$, where -1 represents a signer that believes the certificate is not authentic. Starting from these values, the authors provide the necessary formulas evaluating a user's feedback. Other existing works proposes the usage of blockchain to secure Trust Management system for authentication. For example, in [2] the authors formally model such systems as trust graphs and explore how the usage of a blockchain can mitigate attacks. In [18] the authors proposed a formula to calculate the trust degrees between two users in an e-commerce as a combination of direct and indirect trust degrees. Starting from this formula, the authors in [6] exploits bolckchain as a mean to store the necessary information. The work in [22] proposes a framework supporting fast propagation of certificate revocation and elimination of man-in-the-middle risk by using blockchain. With respect to all these works, our proposals exploit blockchain to store certificate signatures and smart contract to verify the presence of trust chains enabling propagation in Web of Trust. The proposed solution also contrasts the malicious de-anonymization of social identities, in fact the trust propagation stores only information about Ethereum addresses. Moreover, the introduction of a smart contract does not ask clients to be full nodes (i.e., a node that stores the full blockchain and participates in block validation) and to perform expensive computations to navigate the trust graph. The smart contract also introduces economic disincentives to malicious behaviours (such as Denial of Service and Sybil attacks). In the next section, we describe the details of our Ethereum-based proposal for enabling propagation in Web of Trust.

2.2 Ethereum

Since we need a trusted and decentralised mechanism to overcome these issues, we decide to implement a blockchain-based solution enabling the propagation in Web of Trust. In particular, we use a platform based on blockchain called Ethereum [5, 9, 20], which allows the development of DApps (Decentralised applications) that requires to interact each others in a secure and fast way. Ethereum utilises the distributed ledger model by purposing it to model a virtual computer. Indeed, it provides a decentralised virtual machine (Ethereum Virtual Machine - EVM) capable of executing code (the so-called Smart Contracts).

In Ethereum, we can distinguish between two kinds of accounts: (1) Externally Owned Accounts (EOAs); (2) Contract Accounts (i.e., Smart Contracts). The former are controlled by private keys, while the latter are controlled by the code of the contract itself. At the moment, the only high-level and Turing-complete programming language that implements the EVM Bytecode is Solidity [8]. This programming language is used to write and develop Smart Contracts.

As we just said, Ethereum Smart Contracts are real nodes of the network like EOAs and are used as agreements between users who do not trust each other. Indeed, they exploit the decentralised and distributed consensus mechanism of Blockchain that do not requires any Trusted Third Party (TTP).

More in detail, this mechanism is established by mining based on the proof-of-work (PoW) scheme. The PoW assumes that the winner miner is that one who solves first some mathematical puzzles. The average time for mining a block of transactions is about $10 - 12$ seconds. As a consequence, new branches of the principal chain are generated very often and it is necessary to manage these forks to guarantee a certain level of security and decentralisation of the mining process [10]. For this purpose, Ethereum implements a simplified and modified version of the protocol *GHOST* (Greedy Heaviest Observed Subtree) in such a way also “uncles” nodes are partially considered in the computation of which block has the largest and heaviest total proof-of-work backing it.

Another feature of smart contracts is that on of accessing easily data that is stored on Ethereum. Moreover, they can also process, edit or write new data. Finally, it is important to underline that the code of smart contracts can be executed by every node of the blockchain.

One of the most relevant and successful property of Ethereum regards tokens. In this environment, a token is a particular cryptocurrency that has no value until someone or something (e.g. the crypto-market) gives it to it. Usually a company or a single-person that decides to implement a new token via smart contract publishes the business idea in a white paper and offers the token during a Initial Coin Offering (ICO) period [11].

3 DESCRIPTION OF OUR PROPOSAL

The core of our proposal resides in the representation of a Web-of-Trust-based trust model, allowing the propagation of trust among a domain of public key associated with real-life identities. Even if we assume that real-life identities operate on an OSN context, the proposed solution is general enough to be applied in other contexts. The user needs to publish their public keys (e.g., on their own social network profiles). We remark that the focus of this proposal does not regard the problem of impersonation and fake profiles in social networks, for which a wide literature exists, and the existing approaches and techniques can be orthogonally applied together with our solution.

According to the Web of Trust model, every user u elects a number of *introducers*, who are persons *objectively* trusted for u. From the point of view of trust propagation, public keys (actually, certificates) associated with the introducers play the role of *root certificates* whenever u wants to verify trust paths. More formally, we define a set of users U and a function $f_i : U \rightarrow 2^U$, which, for any user $u \in U$ returns the set $f_i(u) \in 2^U$ of the *introducers* of u. Given an user u, we denote by C_u the certificate including the public key associated with the social-network identity of u.

As highlighted in Section 2, in order to avoid the necessity of single-organization trusted services implementing trust propagation and certificate revocation, we leverage Ethereum. Thus, we refer to another domain of identities, which is composed of the set of Ethereum addresses. We require that any user u, to participate in the Web of Trust, is able to associate her/his certificate C_u with an Ethereum address ETH_u. The idea is that the trust graph is built via an Ethereum smart contract *SC*, it is stored also into the state of *SC* and it is managed through the functions of the same smart contract. Let denote by ETH_{SC} the Ethereum address of *SC*. The smart contract *SC* includes the following functions: *sign*, $verify$, and *revoke*. The exact definition of these functions will be explained throughout this section. The status of *SC* is composed of a directed graph G_s of Ethereum addresses, a list L_r of revoked Ethereum addresses, and a data structure storing the number of failing attempts of Ethereum addresses if not null (this point will be explained in the Trust Verification process below).

We represent Ethereum transactions as tuples ⟨*src_address*, *recipient_address*, *data*⟩, where *src_address* denotes the Ethereum address of the sender, *recipient_address* denotes the Ethereum address of the receiver, and *data* is the field including additional information (allowed in Ethereum). In our representation, we do not make explicit the fact that any transaction is signed by the sender by means the Ethereum private key, and we omit some information related to specific features of Ethereum (e.g. GAS price). We highlight that we do not define a generic representation of smart-contract events because the structure of any event can be defined by the smart-contract designer in terms of both structure and content. In the following, we list the content of events into tuples.

Now, we describe how we map, in our model, the basic functions of Web of Trust, which are *certificate signature*, *trust verification*, and *certificate revocation*. We want to highlight that each of the following operations calling the smart contract can be performed only by those Ethereum addresses that have not been revoked yet since the smart contract will filter all the illegitimate requests received.

For the sake of clarity, when we say that a user u *trusts* another user v, we mean that u obtains by our trust infrastructure the information that the user v (actually, her/his certificate C_v) is reliable. According to Web of Trust, we consider three different levels for trust, that are COMPLETE , MARGINAL , UNDEFINED , for decreasing reliability. To be realistic, we assume that users trust only reciprocally because we consider that trust is required as a preliminary step of an interaction between two users who do not know each other. So, when a user u signs a certificate C_v, we say that u *gives trust* to v.

Certificate Signature. This phase is carried out when a user u wants to sign the certificate C_v of another user v. She/he has to generates an Ethereum transaction $T_s = \langle ETH_u, ETH_{SC}, ETH_v \rangle$ calling the function *sign* of the smart contract *SC*, whose effect is to update the status *SC* by inserting the arc (ETH_u, ETH_v) in G_s, provided that ETH_v is not in L_r. Observe that, as said before, this operation can be carried out only by those Ethereum addresses who are not in the revocation list L_r.

Trust Verification. The goal of this process is to obtain that the users u and v know the reciprocal Ethereum addresses and, at the same time, trust each other. Observe that the second result is reached only if, once the Ethereum addresses have been disclosed, the smart contract verifies that the trust paths starting from those addresses satisfy the policies required by the users.

To discourage malicious attempts of users aimed to only discover the association between a social profile and the corresponding Ethereum address, this procedure can be started only by users who demonstrate to own a sufficient amount of trust k (in terms of number of signatures of their certificate). This measure resumes in some sense the *Proof of Stake* protocol [19] In our case, malicious behavior is prevented because the user *risks* her/his trust. Indeed, the smart contract will revoke Ethereum addresses after a certain number n of *failures*. For us, an user u *fails* when she/he does not satisfy the policy P_v of the user v.

So, in this trust verification process, u and v send the following transactions to *SC*. u sends the transaction $T_u = \langle ETH_u, ETH_{SC}, Data_u \rangle$, where $Data_u$ is the following tuple: $\langle f_i(u), R_u, R, P_u \rangle$, in which R_u denotes the result of the encryption with the public key included in C_v and the signature with the private key of u (thus associated with to the public key included in C_u) of a random number R exchanged previously by social-network (out-of-band) interaction between u and v, R is the random in clear text (required to link this transaction with the other one generated by v), and P_u is the policy required by u such that v can be considered trusted.

In turn, v sends the transaction $T_v = \langle ETH_v, ETH_{SC}, Data_v \rangle$, where $Data_v$ is the following tuple: $\langle f_i(v), R_v, R, P_v \rangle$, where where R_v denotes the result of the encryption with the public key included in C_u and the signature with the private key of v (thus associated with to the public key included in C_v) of the same random number R exchanged previously by social-network (out-of-band) interaction between u and v, R, again, is the random in clear text (required to link this transaction with the other one generated by u), and P_v is the policy required by v such that u can be considered trusted.

More in detail, R_u and R_v represent the challenges aimed to prove that the Ethereum addresses ETH_u and ETH_v are owned by u and v, respectively. The signature guarantees the (source) authentication of the challenge, while the encryption of R guarantees the confidentiality of the link between Ethereum address and social network profile.

The effect of the above transactions (for any u) is to call the function $verify$ of *SC* that works as follow:

- first, it checks that the sender u is not in L_r;
- if u is not revoked, then it verifies that C_u has been signed for a number of times greater than the threshold;
- if yes, the function stores the mapping between ETH_u and R;
- for every couple (ETH_u, ETH_v) of ethereum addresses mapped with R, the function checks for trust paths, starting from the ethereum address of the other one v, and computes and returns the *trust* level based on $f_i(u)$ and on the policy required by u (see Section 4 for further details) by emitting an event only in case of satisfaction of such policy;
- indeed, if the policy of u has not been satisfied, then the smart contract updates its status by increasing the value of failures of v.

In particular, events have the following structure : $\langle ETH_u, ETH_v, R_v, R, T_{uv} \rangle$, where ETH_u and ETH_v represent the couple of users u and v linked by R, and T_{uv} is the trust, computed by the smart contract, that u has with respect to v.

Now, each user who called the function listens on the blockchain for events having her/his R. When she/he finds it (or them), u computes the decryption of R_v in such a way she/he can have confirmation that v is actually the one with whom she/he is interacting on the social network.

In our architecture, it is possible that a malicious user z may try to carry out a man-in-the-middle attack since the random R is clearly shown. For mitigating this risk, we add some disincentives in our model. In fact, if it is real that z can generate a transaction with R in such a way the smart contract finds the couples of users (u, z) and (v, z) as well as the legitimate couple (u, v), it is real also that:

- the attacker z must have a corresponding certificate C_z signed at least k times,
- if the attacker z does not satisfy the policy of the victim, the smart contract will update its status in terms of number of failures attempts of z and that, when this counter will be greater than n, her/his certificate C_z will be automatically revoked by adding ETH_z in L_r;
- in Ethereum, every operation costs some amount of gas, so z would spend gas each time she/he will try to carry out an attack.

The result of these countermeasures and precautions is that the attacker z has lost all her/his trust and that her/his ethereum address will be filtered as black listed.

Certificate Revocation. When a user u wants to revoke her/his certificate, she/he has to generate a transaction to SC from her/his ethereum address ETH_u by calling the function *revoke*, which changes the status of the smart contract by adding ETH_u to the list of revoked ethereum addresses L_r. From this moment on, every trust path that passes through ETH_u will be considered as invalid and, moreover, if another user w will have ETH_u in her/his $f_i(w)$, the smart contract will filter this list of introducer by removing ETH_u (and possibly others revoked).

Furthermore, as we said before, there is the case in which Certificate Revocation is carried out automatically by the smart contract to prevent DoS, replay attacks, and so on. In particular, this Certificate Revocation happens when a user fails the *Trust Verification* phase more than n times.

4 IMPLEMENTATION ISSUES

After seeing the description of our proposal, let's move on to some implementation details. First, in Section 3, we introduced the policy P_u of the user u that must be satisfied in order to proceed with the event emission from the function `verify` of the smart contract SC. In particular, with P_u, we intend a function based on the two well-known parameters of Web of Trust *(i)* `COMPLETES_NEEDED` and `MARGINALS_NEEDED` [1]. These two parameters work as thresholds, in the sense that they define the number of full trusted introducers or marginal trusted introducers needed to reach the desirable trustworthiness of the certificate. More in detail, Figure 2 depicts the declaration of the function `verify` which corresponds to the $Data_u$ field described in Section 3.

```
pragma solidity 0.5.7;

contract SC {
    ...
    function verify(address _to, address[] _introducers, bytes32
         ciphered_random, uint256 random, uint256 completes_needed
         , uint256 marginals_needed){
    ...
    }
}
```

Figure 2: Declaration of the function `verify`

Furthermore, it is important to give more details about how we effectively propagate trust. As said before, we want to remark that trust should be considered more objective than it is, because if I give trust to another person, then I am giving trust to her/his decisions too.

Anyway, it is also realistic to think that, during propagation, the trust value should decrease after a certain number of hops. Moreover, even if we implement a smart way to store and manage the trust graph in the smart contract, since every operation carried out by it is onerous, we apply the theory of the *small world* and *six degrees of separations* [14, 17] and of the so-called *horizon of observability*, which consists of a value, deriving from network theory which is, in turn, related to the FOAF (Friends of a Friend) concept, that oscillates between two and three [12] in the following way:

- if the hop counter needed to reach my introducers is less than the horizon of observability (that is equal to 3), then the trust level propagates without any decreasing;
- instead, if the hop counter needed to reach my introducer is greater than 3, the trust value decrease (from *full* to *marginal* and from *marginal* to *don't know*);
- in particular, when the hop counter reaches a value equal to six (like the degrees of separations), the algorithm stops in order to avoid to spend too much gas.

```
pragma solidity 0.5.7;

contract SC {
    ...
    event trust_satisfied(address indexed _from, address indexed
         _to, bytes32 ciphered_random, uint256 indexed random,
         string t_value);
    ...
    }
}
```

Figure 3: Declaration of the event `trust_satisfied`

After explaining how we propagate trust, let's move on the emission of the event. As shown in Figure 3, we called the event `trust_satisfied` and it has all those parameters necessary to communicate that the phase verification of the trust has been successful. Observe that, the keyword `indexed` allows filtering queries on all events logged by the smart contract with respect to those parameters preceded by this keyword.

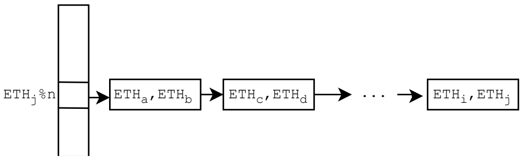


Figure 4: Store schema

To store the information about the trust graph, the smart contract internally has a high storage capacity but it has to organize data in a static array. Figure 4 depicts the data structure used by the smart contract for the trust graph. A static array with a length n is used (left of the figure). We associate a list of blocks, each one containing a couple of Ethereum addresses, to each element of this array. When the smart contract has to store the information about a trust from the Ethereum address ETH_i to ETH_j, the smart contract enters the list of blocks addressed by the mod (modulo) operation on the address ETH_j and appends the new block made of the couple $< ETH_i, ETH_j >$ to the list. Intuitively, a high value for the dimension n reduces the number of collisions among Ethereum addresses, increasing the needed data space. This data structure represents also an efficient system in which to search for the addresses signing an Ethereum address ETH_i. The smart contract has to scan the list associated with the i mod n location and search for ETH_i in the second element of the blocks.

The same data structure is also used inside the smart contract to store both the revocation list L_r and the number of failing attempts of Ethereum addresses. For both these cases, the key to access the array is represented by the Ethereum address to store mod$n$, while the blocks are structured as a couple of Ethereum address and a boolean flag for the revocation list, Ethereum address and a counter for the number of failing attempts.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a solution, based on Ethereum and smart contracts, to enable propagation in Web of Trust. In particular, our solution exploits Ethereum to avoid the necessity of single-organization trusted services. Moreover, the privacy of users is protected since the proposal uses Ethereum addresses to store trust relationships. Strong measures are also proposed to contrast malicious de-anonymization of the couples addresses-identity. A smart contract stores the current status of certificate signatures and manages revocations. The usage of a smart contract simplifies the client operations, that do not perform expensive computations to navigate the trust graph. A smart contract represents also an economic disincentive to malicious behaviours.

As future work, we plan to apply our solution in different domains in which trust propagation is required. Furthermore, we will apply formal verification techniques to formally prove the security level of our solution. At last, we will try to adopt more effective challenges to prove the ownership of Ethereum addresses without the possibility of the social identity disclosure.

REFERENCES

[1] Alfarez Abdul-Rahman. 1997. The PGP Trust Model. In *EDI-Forum: the Journal of Electronic Commerce*, Vol. 10. 27–31.

[2] Nikolaos Alexopoulos, Jörg Daubert, Max Mühlhäuser, and Sheikh Mahbub Habib. 2017. Beyond the hype: On using blockchains in trust management for authentication. In *2017 IEEE Trustcom/BigDataSE/ICESS*. IEEE, 546–553.

[3] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. 2009. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*. ACM, 551–560.

[4] Francesco Buccafurri, Gianluca Lax, Denis Migdal, Serena Nicolazzo, Antonino Nocera, and Christophe Rosenberger. 2017. Contrasting False Identities in Social Networks by Trust Chains and Biometric Reinforcement. In *2017 International Conference on Cyberworlds (CW)*. IEEE, 17–24.

[5] Vitalik Buterin et al. 2013. Ethereum white paper. *GitHub repository* (2013), 22–23.

[6] Kun-Tai Chan, Raylin Tso, Chien-Ming Chen, and Mu-En Wu. 2017. Reputation-Based Trust Evaluation Mechanism for Decentralized Environments and Its Applications Based on Smart Contracts. In *Advances in Computer Science and Ubiquitous Computing*. Springer, 310–314.

[7] Mauro Conti, Radha Poovendran, and Marco Secchiero. 2012. Fakebook: Detecting fake profiles in on-line social networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. IEEE Computer Society, 1071–1078.

[8] Chris Dannen. 2017. *Introducing Ethereum and Solidity*. Springer.

[9] ethereumWiki. 2016. Ethereum project. https://github.com/ethereum/wiki/wiki. (2016).

[10] ethereumWiki. 2019. Ethereum White Paper. https://github.com/ethereum/wiki/wiki/White-Paper. (2019).

[11] Gianni Fenu, Lodovica Marchesi, Michele Marchesi, and Roberto Tonelli. 2018. The ICO phenomenon and its relationships with ethereum smart contract environment. In *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. IEEE, 26–32.

[12] Noah E Friedkin. 1983. Horizons of observability and limits of informal control in organizations. *Social Forces* 62, 1 (1983), 54–77.

[13] Simson Garfinkel. 1995. *PGP: pretty good privacy*. " O'Reilly Media, Inc.".

[14] John Guare. 1990. *Six degrees of separation: A play*. Vintage.

[15] Guibing Guo, Jie Zhang, and Julita Vassileva. 2011. Improving PGP web of trust through the expansion of trusted neighborhood. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 489–494.

[16] Rolf Haenni and Jacek Jonczy. 2007. A New Approach to PGPâĂŹs Web of Trust. In *EEMAâĂŽ07, European e-Identity Conference*.

[17] Stanley Milgram. 1967. The small world problem. *Psychology today* 2, 1 (1967), 60–67.

[18] Xiu-Quan Qiao, Chun Yang, Xiao-Feng Li, and Jun-Liang Chen. 2011. A trust calculating algorithm based on social networking service users' context. *Jisuanji Xuebao(Chinese Journal of Computers)* 34, 12 (2011), 2403–2413.

[19] Pavel Vasin. 2014. BlackcoinâĂŹs proof-of-stake protocol v2. *URL: https://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper. pdf* 71 (2014).

[20] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151 (2014), 1–32.

[21] Cao Xiao, David Mandell Freeman, and Theodore Hwa. 2015. Detecting clusters of fake accounts in online social networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*. ACM, 91–101.

[22] Alexander Yakubov, Wazen Shbair, and Radu State. 2018. BlockPGP: A Blockchain-based Framework for PGP Key Servers. In *2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW)*. IEEE, 316–322.

[23] Philip R Zimmermann. 1995. *The official PGP user's guide*. MIT press.

Meta-Data Management and Quality Control for the Medical Informatics Platform

Admir Demiraj
ntemirai16@aueb.gr
Department Of Informatics
Athens University of Economics and Business
Athens, Greece

Kostis Karozos
karozos@aueb.gr
Department Of Informatics
Athens University of Economics and Business
Athens, Greece

Iosif Spartalis
isparta@aueb.gr
Department Of Informatics
Athens University of Economics and Business
Athens, Greece

Vasilis Vassalos
vassalos@aueb.gr
Department Of Informatics
Athens University of Economics and Business
Athens, Greece

ABSTRACT

The Medical Informatics Platform (MIP) of the Human Brain Project (HBP) is tasked with providing its users diverse high quality clinical data and tools for medical analysis, while complying with the national legislation about privacy and security. Data, which is provided by a large number of hospitals, tends to be heterogeneous and also has a constantly changing schema, due to hospitals' need to capture more information. In this paper we provide a look in the MIP's data ingestion pipeline and focus on steps taken by our team to properly integrate clinical data from heterogeneous sources while ensuring its quality throughout the processing pipeline. We have developed tools both for meta-data management and quality control.

CCS CONCEPTS

• **Information systems → Version management**; **Information systems applications**; Data management systems; Information integration.

KEYWORDS

Database Management, Meta-Data Management, Quality Control, Schema Matching, Data Integration, Clinical Data, MIP, Medical Informatics Platform

ACM Reference Format:
Admir Demiraj, Kostis Karozos, Iosif Spartalis, and Vasilis Vassalos. 2019. Meta-Data Management and Quality Control for the Medical Informatics Platform. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3331076.3331088

1 INTRODUCTION

There is a common understanding the human brain has been one of the most compelling and difficult tasks that humanity has undertaken. The possible implications from such an endeavour are countless and would have a huge impact on many aspects of our society. The HBP [2] is a large 10-year scientific research project, that aims to create an infrastructure of cutting edge technology, in order to allow researchers to advance the fields of brain-related medicine, computing and neuroscience. In order to achieve such an ambitious goal six Information and Communication Technology (ICT) platforms have been created: the Neuroinformatics Platform[1], the Brain Simulation Platform[2], the High-performance Analytics and Computing Platform[3], the Medical Informatics Platform[4] [2], the Neuromorphic Computing Platform[5] [5] and the Neurorobotics Platform[6] [7].

For this paper we will be focusing on the Medical Informatics Platform (MIP), that aims at understanding disease clusters and their respective disease signatures through local and federated analysis of data, residing in a wide network of hospitals. The federated analysis is a crucial component of the project as it combines analytical results from data stored in various sources, while guaranteeing that no sensitive information is leaked out of the facilities of the hospitals. By overcoming the issues of privacy preservation that are imposed by national legislation and institutional ethics, the project intents to encourage the scientific community to experiment with records that until recently had been totally inaccessible.

In the medical field, providing high quality information is of utmost importance in order to have a valid analysis. This task becomes especially difficult when considering that data resides in multiple sources and is heterogeneous, containing both electronic health records (EHR) and imaging features. In order to make the analysis feasible across all hospitals, data that is provided has to be matched

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331088

[1] https://www.humanbrainproject.eu/en/explore-the-brain/neuroinformatics-platform/

[2] https://www.humanbrainproject.eu/en/brain-simulation/brain-simulation-platform/

[3] https://www.humanbrainproject.eu/en/hbp-platforms/hpac-platform/

[4] https://www.humanbrainproject.eu/en/medicine/medical-informatics-platform/

[5] https://www.humanbrainproject.eu/en/silicon-brains/neuromorphic-computing-platform/

[6] https://www.humanbrainproject.eu/en/robots/

to a unified global schema. We will be referring to the elements of the global schema as Common Data Elements(CDEs[7]). The CDEs, which describe the knowledge about patients' brain structure, diagnosis, clinical tests results as well as genetic and demographic information in a clear and straightforward way, are a product of HBP's clinicians' and researchers' scientific work. The process of aligning the local variables to the CDEs is usually referred as a mapping task and is done with the use of MIPMap [11] [21], a schema mapping and data exchange tool. In many cases the schema of the hospital variables changes due to errors or empty values or simply because we want to incorporate more information. Moreover, the global schema is also updated because as more hospitals are providing their data we find out that we need to extend it to contain more elements that were not encountered before and are present in many new hospitals. Concluding, this means that we need to map the changing schemas of variables from different hospitals, to a changing global schema and each change has to be communicated to a large team. This complication led soon to the need to create a single point of reference for maintaining and implementing changes to the schema of both the variables and the CDEs. Another issue we encounter is that in many cases data that is provided by the hospitals is of poor quality and we have to take the decision of either correcting it or dropping it entirely. This means that we need a way to identify which records might be faulty and act accordingly. In this paper we present both a tool for managing the schemas of the data as well as a tool that provides a quality analysis for the actual data. We will be referring to the first tool interchangeably as Meta-Data Management Tool (MDMT) or Data Catalogue (DC) and to the second tool as Quality Control Tool (QCT).

2 RELATED WORK

2.1 Meta-Data Management

Sen and Arun [18] describe the growing implications of meta-data in multiple fields. In their work Lauren E. Sweet and Heather Lea Moulaison [20] denote the importance meta-data interoperability for a wider medical analysis and argue that there are a number of issues in applying a common standard due to the complexity of the data. Greenberg et al. [8] note that data is created faster than meta-data can organize it. Curdt et al. [6] describe their experience in creating a meta-data management tool and argue about the importance of versioning. Jiang et al. [9] provide an overview on an end-to-end meta-data management solution.

2.2 Quality Control

Adlassnig et al. [1] and Shabestarial et al. [19] provide an overview on the requirements and challenges of quality assurance in EHR. Perimal-Lewis et al. [16] try to identify the key factors that influence and determine the quality in EHR. Kerr et al.[12] show the best practices and potential benefits in the management of quality in EHR. Boyle et al. [4] created a system to evaluate the quality of EHR by identifying the latest valid results and the data sources that they were extracted from. Orfanidis et al. [15] focus on the quality issues that are encountered in the adaptation of EHR for managing the data in medical facilities. They also propose a system that will implement sequential steps in order to ensure quality. Arts et al. [3] have identified through research several quality procedures for the medical registries and created a framework for their implementation.
Prieto et al. [17] utilize the DICOM header of radiology images to identify slices that were retaken and thus eliminating duplicate information. Kallman et al. [10] identify the changes in the patients' exposure and in the quality of the image by using an automated method to analyze data from DICOM headers. Liu et al. [13] have developed a quality assurance program that evaluates radiology interpretations based on reference standards. They create performance metrics for their evaluation and compare them against other benchmarks.

3 DATA INGESTION PIPELINE IN THE MIP

3.1 MIP architecture overview

A high level overview of MIP would reveal four basic layers. Following a top-down approach the first layer would be the MIP Portal, which provides the medical researcher with a set of tools for analysis over the data of the hospitals that has been mapped to the CDEs. The second layer is the Federation layer which is responsible for taking the requests for analysis from the Portal and executing them over multiple hospitals. Moreover, this layer is responsible for retrieving the output of the analysis from each hospital and aggregating the result that will ultimately be returned to the Portal.

The third component is the Local layer, which is located in the facilities of each hospital and runs analysis only on local data. Finally, at the bottom of the architecture lies MIP's Data Factory, which is responsible for providing the platform with data. Our contribution is focusing on the procedures ran by the Data Factory that we describe in depth in the next section.

3.2 MIP Data Factory

In (Figure 1) we present the architecture of the Data Factory. On the far left side we can see the hospital that collects all the available information from various sources (Inbound Information Systems, REDCap[8]) and provides three types of data. The first one is Brain Scans, that are being produced by Magnetic Resonance Imaging (MRI) machines and are provided in accordance with the DICOM[9] [14] integration standard. The second type is the Electronic Health Records (EHR), which are basically all the information a hospital collects from its patients. EHR may include, but are not limited to, demographics, medical history, medication and allergies, immunization status and laboratory test results. These records have both a vast range as well as massive differentiation from hospital to hospital. Some of the reasons behind this are that the hospitals may differ in the equipment that they have available, the laws and legislation that apply in the region and the information systems that they are being used for storage. Finally, the last type of data is the meta-data, that describe the schema of the EHR. The meta-data are provided in a standardized format that was created for the needs of the project. Despite the fact that the data will be accessed within the environment that is provided by the hospitals, an additional level of security is employed through the process of Pseudonymization.

[7] the current schema of the CDE's contains 172 elements and is published in: https://github.com/HBPMedical/mip-cde-meta-db-setup/blob/master/variables.json

[8] https://www.project-redcap.org/

[9] https://www.dicomstandard.org/

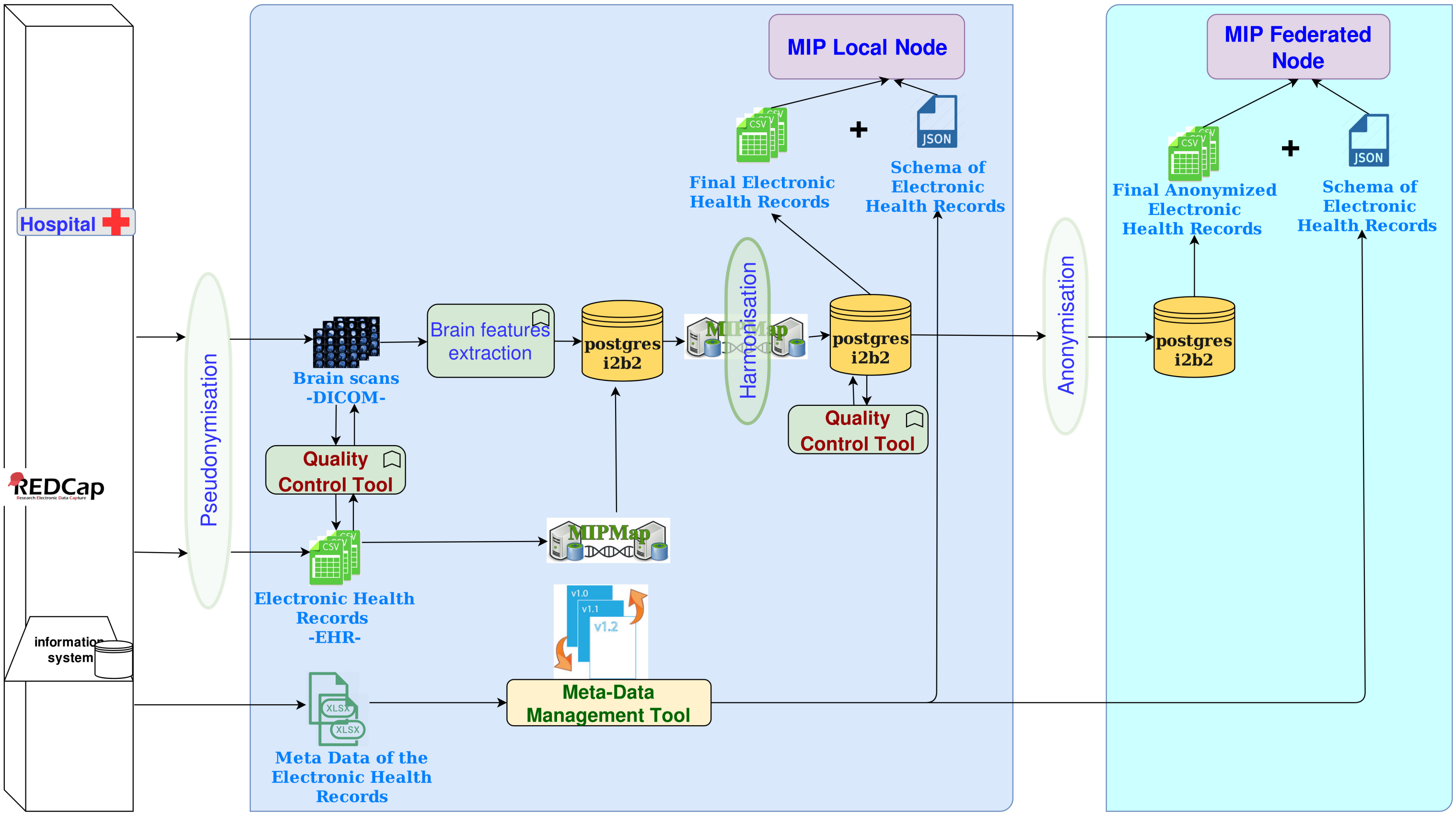

Hospital
REDCap
information system
Pseudonymisation
Brain scans -DICOM-
Brain features extraction
postgres i2b2
Quality Control Tool
Electronic Health Records -EHR-
MIPMap
Harmonisation
postgres i2b2
Quality Control Tool
MIP Local Node
Final Electronic Health Records
+
Schema of Electronic Health Records
Meta Data of the Electronic Health Records
Meta-Data Management Tool
v1.0
v1.1
v1.2
Anonymisation
postgres i2b2
MIP Federated Node
Final Anonymized Electronic Health Records
Schema of Electronic Health Records
CSV
JSON
XLSX

Figure 1: The Data Factory of the Medical Informatics Platform.

The hospital personnel decides in accordance with each hospital's policy which record columns should be eliminated of replaced with other values (e.g. patient name and patient id) so as the data to be depersonalized.

The brain scans are passed through a pipeline, that extracts brain morphometric features. As for the EHR the initial step involves a schema matching process to fit in the tables, that we have defined in the first Postgres database (unharmonized). All such processes are implemented using MIPMap. MIPMap offers a GUI that allows the user to define correspondences between the source and the target schema, as well as join conditions, possible constraints and functional transformations. Users can simply draw arrow lines between the elements of two tree-form representations of the schema. The techniques employed for automated or semi-automated schema matching are prone to mistakes that due to the nature of the biomedical field we cannot afford. We are processing sensitive information and we have to guarantee their integrity. After the mapping the EHR are linked with the Brain Scans and are now ready for the harmonization process. The harmonization process involves mapping the first Postgres database's variables to the CDEs while processing their values so as to conform to the CDEs standards and measurement units. The finally harmonized data is saved in another database. This harmonized database is essentially feeding the MIP local node with data. These nodes combine the CDEs with their schema, that is being offered by the MDMT, to provide data for analysis to the upper layers of MIP. The local node differs from the federated one in the manner that it is not contained in the federated analysis. Depending on their access rights, users can access the local node and run analysis only on the data of a specific hospital. The tools and algorithms the user can use in such an analysis are different from the ones he would for federated analysis. For security purposes an anonymization process is followed in the records that will be included in the federated analysis.

In this pipeline in order to ensure the quality of the given data the QCT is being deployed iteratively both in the initial EHR data and brain scans and in the final harmonized database. The reports it produces are being used for the decision to either correct or delete certain variables. The reports are saved and can be referenced though the MDMT. The MDMT is responsible for taking the initial meta-data and producing a final JSON that contains a harmonized schema of the variables.

4 META-DATA MANAGEMENT

4.1 The importance of Meta-Data

The most common description of the meta-data is "data about the data". We could more accurately depict it as the information required to contextualize and understand a specific data element. In this paper we will be focusing on meta-data describing clinical data.

Nowadays hospitals and medical facilities collect a vast amount of information by using a variety of techniques and tools. In order to have an accurate depiction of this information meta-data is essential. Meta-data is being utilized in the vast majority of medical surveys and one could argue that it is as important as the actual data. The case becomes even stronger when considering the need to combine data from multiple sources for distributed analysis. There is a number of hospitals all over Europe contributing in the HBP and as it is expected, there are more than a few differences on which variables and meta-data each one is collecting and in which format. Thus, we need a standardized way of collecting meta-data, in order not only to understand the given variables, but also to be able to efficiently compare them. One of the main tasks of the MIP is mapping hospital data to a global schema. The task is inaccurate and time-consuming without a good understanding of the clinical data. The margins for errors are slim and we need to be able to guarantee high quality standards. As an added benefit, the meta-data act as our first line of defence against any inaccuracies in the given variables. A simple comparison of the value of the data with the given meta-data specifications can prevent a number of mistakes from propagating to later stages of the data pipeline.

4.2 Meta-Data Management Tool

General Description: The MDMT is an end-to-end platform, created with the latest technologies and tasked with versioning the meta-data of the hospital variables and the CDEs. It utilizes a global schema for the collection of the meta-data and provides information about the mappings of the variables to CDEs. The potential users of the tool are first of all the researchers that prior to executing experiments through the Portal of the MIP may want to investigate what type of information is available in each hospital. Moreover, the tool aims at facilitating the collaboration between the authorized hospital personnel and the development team of MIP. Each side can implement changes by creating a new version, which will be reviewed by the other side and through many iterations, this process will lead to high quality information. As a result we will be able to make more accurate mappings of variables to CDEs and resolve cases where we did not have enough information to make a mapping. The data pipeline of the MIP is quite big and a more than one team is involved in each process, so when a change is made everyone has to be informed. The tool is intended to be a single point of reference and hopefully will eliminate any mistakes originating from the lack of transparency in the changes implemented in the variables' schema and meta-data.
Changes in the schema of clinical data are quite common and we can distinguish three types of them:

(1) Correction changes due to mistakes or to insufficient information in the meta-data.
(2) Changes due to information coming from a new data source within the hospital.
(3) Changes due to updating the CDEs and thus new mappings are possible or old ones might have to be modified.

At any given time we can download the meta-data for the MIP's global schema as well as the meta-data for every hospital's schema, of imported (or to-be-imported) data. All these are stored in JSON files. These JSON meta-data files are used as input to the MIP when importing new data, since the platform along with the data is in need of its corresponding meta-data.

Meta-Data Information: The process of tracking the meta-data is initiated by each hospital that has to upload, in a specified format, an excel file containing all the meta-data. We have defined a set of meta-data variables for the hospitals in order to avoid confusions and to speed up the process of uploading them. This set contains the following information:

- **csvFile:** The name of the dataset file that contains the variable.
- **name:** The name of the variable.
- **code:** The variable's code.
- **type:** The variable's type.
- **values:** The variable's values. It may have an enumeration or a range of values.
- **unit:** The variable's measurement unit.
- **canBeNull:** Whether the variable is allowed to be null or not.
- **description:** The variable's description.
- **comments:** Comments about the variable's semantics.
- **conceptPath:** The variable's concept path.
- **methodology:** The methodology the variable has come from.
- **mapFunction:** The function that transforms the variable's value into the value of its corresponding CDE.
- **mapCDE:** The corresponding CDE.

This information covers the vast majority of the meta-data we want to collect and is easy to adopt even by hospitals that have not implemented a standard about their meta-data. Most of the columns are self explanatory but we feel we should give further information about the conceptPath, the mapFunction and the mapCDE. The conceptPath essentially defines the hierarchy of the variable. It has a strictly defined format and always starts with the root category (/root) followed by the rest of the consecutive categories, separated by slash (/) until we reach the variable code. For example the conceptPath for the Mini Mental State Examination[10] score is /root/neuropsychology/minimentalstate. The mapCDE defines to which of the CDEs should the current variable be matched and the rule that should be applied to its value for the transformation is expressed by mapFunction. The tool keeps the information about the mappings but it does not make the actual mapping transformations (this is done by MIPMap). After the provided file is checked for its integrity, a new hospital entity is created containing the first version of the variables' meta-data and mapping.

Meta-Data Viewing: The user can view all versions of each hospital's local variables. She is also able to view all CDEs' versions. For every hospital's meta-data version that is created, we also create a harmonized one containing CDEs and additional hospital local variables. Each meta-data version is depicted via four different views:

(1) **Flat View:** A flat view includes all the variables and their respective meta-data. The variables are searchable by their

[10] The Mini Mental State Examination (MMSE) or Folstein test is a 30-point questionnaire that is used extensively in clinical and research settings to measure cognitive impairment. It is commonly used to screen for dementia.

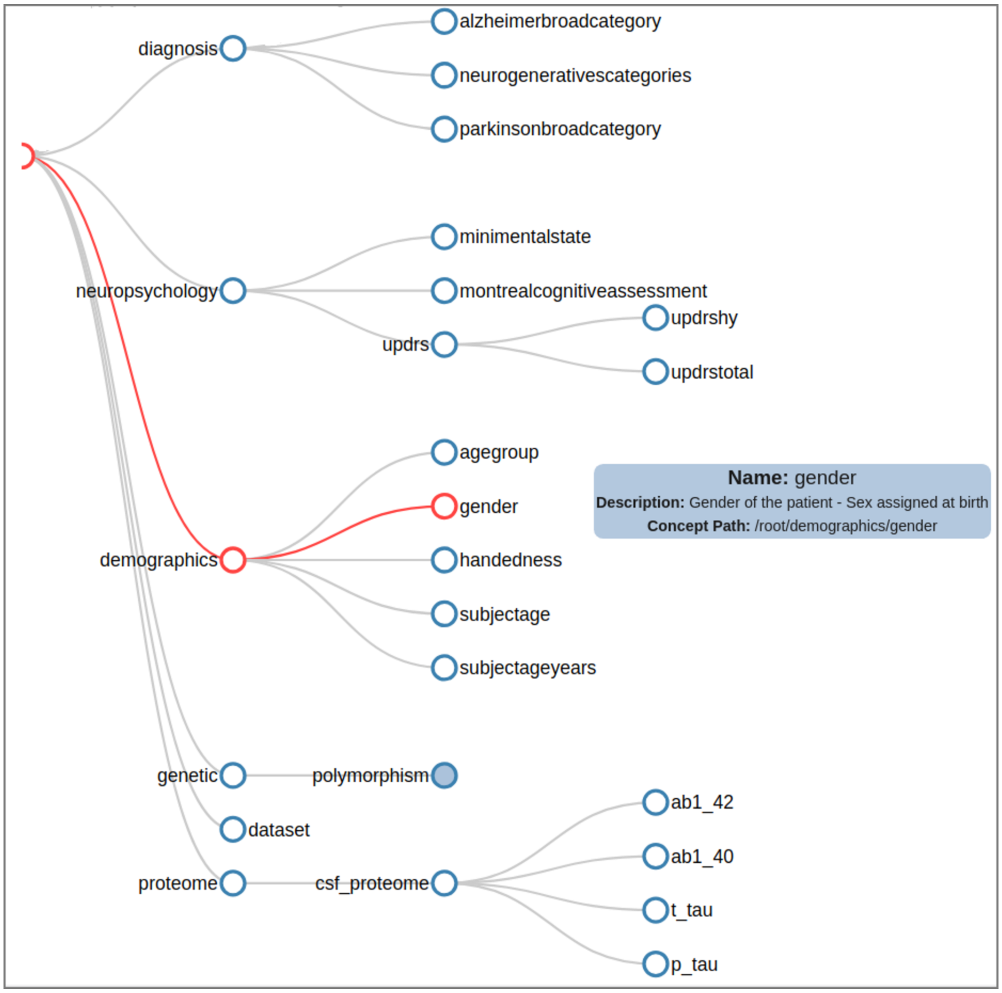


Figure 2: The searchable graph displaying the taxonomy of the variables and CDEs.

code and category. This view is convenient for checking variables' details.

(2) **Tree View:** An interactive tree view (Figure 2) displaying the taxonomy of the variables. The user can expand or collapse nodes and every time she hovers over a node a brief description of the node is displayed. The tree is also searchable by variable code and category. This view is good for a better comprehension of the clinical variables' categories and hierarchical semantic structure.

(3) **Mapping Visual:** For each variable that has been mapped to a CDE, we offer a comprehensive graphical representation (Figure 3) displaying their link and the rule the transformation is made. This view will be later used as a reference to make the actual mapping transformations.

(4) **Quality Control Tool Report:** For each batch of data we receive from a hospital, the QCT produces a report about the whole batch, as well as a report for each of the variables. Both reports are displayed in a table like structure. On this view, we can index each variable to view its report or download it (csv format) for further analysis.

We have to note that for the CDEs only the first two views (flat and tree view) are offered. There is no mapping visual since the CDEs are the schema to map to and we do not produce a quality report for them, because there are no pure CDE datasets. The tree view is an essential component in providing insight on which CDE a hospital's local variable should be mapped to. A user, knowing all the current categories and their hierarchy, can isolate the category that is of interest to her and either find a CDE that the variable can be mapped to or if no CDE is semantically equivalent, she can still place the variable in a new category that is more appropriate, by creating a new leaf node. Furthermore, there are some cases of variables belonging to categories that are not already contained in the hierarchy of the CDEs. In this occasion a new category will be created (intermediate node) for the new version and displayed in the tree. The Data Governance pertinent committee, composed of clinicians and researchers, makes decisions periodically on CDEs' maintenance and enrichment. When new clinical variables and categories see the light, they are considered to be candidates for the CDE-stamp on the next version. To be approved though and become part of the CDEs' global schema they have to have a certain level of commonness between hospitals as well as a scientifically clear and complete definition along with values' range or enumeration. Lastly, they have to be considered having a significant potential for contributing to the MIP analyses' results.

Meta-Data Management: There are two ways a user can create a new version:

1. **Uploading Version File:** The user can upload a file (.xlsx) in a specified format containing the meta-data for all the variables. If the file passes the integrity checks a new version will be created. We also offer prior to the procedure an empty template file containing all the appropriate columns and give detailed instructions on how to complete each one of them. This method is suitable for uploading large amount of information mainly for the creation of the first versions of the variables.
   When parsing a file for the creation of a new meta-data version, DC generates a hierarchical tree out of a flat input, which although being flat has the clinical variables' taxonomy expressed via their conceptPaths. Our system manages to correctly parse the variables of any kind of hierarchy. Every node in the taxonomy may be a category or a leaf (variable). Each row of the file contains the description of either a variable or a category. It is obligatory to give a definition for each different variable, whereas for the categories it is optional. If a category is not defined, we can still create it as long as it is referenced in the hierarchy of its children. The sequence in which the variables or the categories are given in the file is irrespective i.e., children may precede their parents or the other way around. Our parsing algorithm, which uses DFS recursive traversal, will produce the correct tree as long as the given conceptPaths do not contain errors.
   Procedure ADDNODE() is executed for all rows of the input flat file to create an in-memory hierarchical structure of the variables. This variables' tree is serialized for the purposes of meta-data tree visualization as well as data importing as already stated.
2. **New Version GUI:** We also offer a GUI for each version creation. The GUI displays all the variable information of the previous version and gives the ability to either delete or change them as well as add information about a completely new variable. After all the appropriate changes are made the version is submitted to be saved. Given that all integrity tests pass a new version is created. The GUI offers a good and easy way to create minor changes to the already existing version. By constantly making minor corrections to the last version we be gradually creating higher quality information.

Authorization: All types of meta-data searching and viewing are offered to all users without having to login to the platform. In order the user to be authorized to create a new version for hospital data or CDEs she has to login and also have the required rights provided in a centralized way by the Medical Informatics Platform.

Algorithm 1 Creating the variables' tree

```
procedure ADDNODE($row$, $root$)    ▷ add row's variable/category to root's Tree
    $concept[] \leftarrow split(row.conceptPath, "/")$ ▷ split conceptPath into its parts and store them into an array
    for each $concept[i]$ do
        DFS on $root$ for $Node$ having $concept[i]$
        if $Node$ is in $root$'s Tree then
            if $concept[i]$ is the last part of $row.conceptPath$ then
                Update the existing $Node$ with $row$'s metadata    ▷ if a node for $row$ already exists it should have been created when processing one of its children since the current row is the one dedicated to it having all the element's information
        else
            DFS on $root$ for the parent of the $Node$ to add
            if $concept[i]$ is the last part of $row.conceptPath$ then
                Create the $Node$ for $row$
            else
                Create a $Node$ with the up until now conceptPath    ▷ Even though the current row is not dedicated to this element we will create a node for it now. After all, the user is not obligated to give meta-data for the intermediate elements of the tree
            Add the new $Node$
        $root \leftarrow$ the $Node$ for $row$▷ So as not to search from the root of the tree in next concept[i] iteration
procedure CREATETREE($inputFile$)
    Create the $root$ of the Tree
    for each $row$ in $inputFile$ do
        ADDNODE($row$,$root$)
```

Technologies Used: The MDMT is temporally being hosted[11] within the premises of Athens University of Economics and Business (AUEB) and its source code is available at github[12]. The system has been built with the following technologies:

- Java and Spring Boot for the server side application logic.
- PostgreSQL for data storage.
- Angular and TypeScript for client side User Interface (UI).
- D3 for data visualizations.
- Git for source code version control.

5 QUALITY CONTROL TOOL

General Description: The main functionality of the QCT is to ensure a good level of data quality by producing a report (in csv and pdf) describing an incoming hospital dataset before inserting it into the MIP platform and after it has been inserted in the harmonized database. At this stage of development, the tool can process datasets in the form of tabular data (csv) and DICOM datasets containing MRI sequences. The QC tool produces a different report for each kind of dataset which is meant to be evaluated by a human. If the dataset meets the minimum specifications, it is inserted into the MIP.

5.1 Tabular Data

In the case of a tabular dataset, the produced report includes a set of statistics profiling the missing values across the rows (per

[11] http://195.251.252.222:2442/hospitals
[12] https://github.com/HBPMedical/DataCatalogue

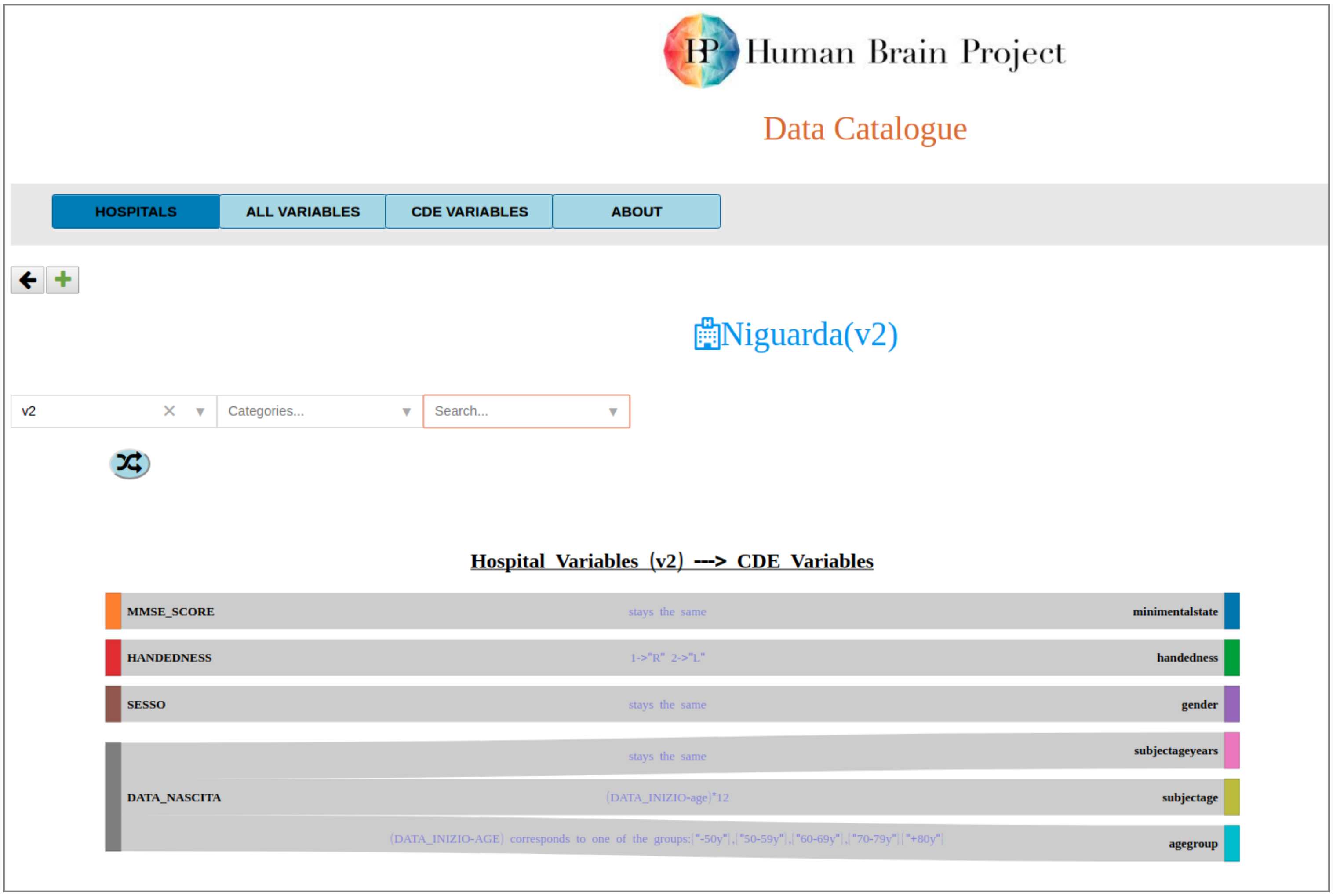


Figure 3: The graphical representation of the variables' mapping to CDEs.

observation) and columns (per variable). Moreover, a set of statistics is calculated per variable depending on the variable type. At the current state, MIP local layer has three types of variables - numerical, nominal and text. For each numerical variable the QCT calculates a set of descriptive statistics - mean, standard deviation, minimum, maximum, 1st, 2nd and 3rd quantile. In addition to that, based on those measurements, the QCT estimates the number of rows with possible outliers per numerical variable. The report for a nominal (categorical) variable provides information about the number of categories and their labels, the most frequent category and the number of occurrences of the latter. Likewise, for a text variable the report provides the same information about the most frequent value, but also includes the 5 most and 5 less frequent values of the text variable. All in all the statistics produced for the variables are the following:

- **Variable Name:** The column name in the given dataset.
- **Type Declared:** The data type of the variable as declared in the file.
- **Type Estimated:** The data type of the variable as estimated by the QCT.
- **List of Category Values(nominal variables):** A string with the category values of the variable (confined in single quotes and separated by commas).
- **Number of Category Values(nominal variables):** The total number of the category values of the variable.
- **Count of Unique Values (text variables):** The count of unique string values.
- **Most Frequent Value (text and nominal variables):** The most frequent category value of the variable.
- **Number of Occurrences for Most Frequent Value (text and nominal variables):** The number of occurrences for most frequent category value of the variable.
- **Count of Records Filled In:** The number of rows that are filled.
- **Percentage of Non Null Rows:** The percentage of filled rows.

- **Mean Value (numerical variables):** The average of the value of the variable.
- **Standard Deviation(numerical variables):** The standard deviation.
- **Minimum Value(numerical variables):** The minimum value of the variable.
- **Maximum Value(numerical variables):** The maximum value of the variable.
- **First Quantile (numerical variables):** The value of the variable that 25% percent of records are below it.
- **Median (numerical variables):** The value of the variable that 50% percent of records are below it.
- **Third Quantile (numerical variables):** The value of the variable that 75% percent of records are below it.
- **Number of Outliers (numerical variables):** The values of the variable that are outside three standard deviations from mean value.
- **The 5 Least Frequent Values (text variables):** A string with the 5 least frequent values of the variable separated by commas.
- **The 5 Most Frequent Values (text variables):** A string with the 5 most frequent values of the variable separated by commas.
- **Comments:** A string with various messages about the the values of the variable.

5.2 Imaging Data

In the case of a DICOM dataset, the QCT has the ability to recognize the MRI sequences in a given folder and extract their meta-data (headers). Based on this meta-data, the QCT performs a validation of the MRI sequences and exports the results in a report. HBP MIP has some specific minimum requirements that every DICOM sequence must meet in order to be inserted to the platform. The requirements that each image should meet are the following:

(1) The images must be full brain scans.
(2) The images must be provided either in DICOM or NIFTI format.
(3) The images must be high-resolution (max. 1.5 mm) T1-weighted sagittal images.
(4) The images must contain at least 40 slices.

If the imaging file is not readable or the requirements are not met the tool produces a report detailing the reasons the file is rejected.

5.3 Technologies Used

The QCT is employed locally within the hospitals and is utilizing the following technologies and libraries:

- Python 3.5 for the development of the main frame.
- Numpy - Pandas to handle the data and produce statistics.
- Pydicom to read DICOM files.
- Tkinter to create the User Interface (UI).

6 FUTURE WORK

Currently we are working towards the improving and robustisation of the Data Factory pipeline so as to guarantee hospital data harmonization and ingestion into the MIP with the less possible human intervention along with high data quality. To that scope we are designing a complementary Quality Control Tool that will be giving recommendations for value corrections that can be useful to hospitals' personnel.

Towards providing more to the MIP user than numeric statistical results, we plan on incorporating some of LORIS' images visualization and quality control features for the clinicians to be able to view in 3D the collected brain scans. Due to the brain scans being data of high sensitivity, to comply to all privacy rules this will be a feature accessed only in MIP Local by a few authorised users. Regulations not allowing exposing raw data to the federation which can be used for personalizing the data (link the information to the actual patient) are an obstacle to giving clinicians access to a large corpus of actual brain scans collected from different hospitals.

7 CONCLUSION

The data pipeline of MIP is complex and demands high quality guarantees due to the sensitive nature of the information. The schema of the variables and the CDEs is constantly changing due to ever growing need to capture more information and due to corrections. The need of understanding the hospital variables is essential for the process of mapping them to CDEs. We are presenting an end-to-end tool for meta-data management that offers an efficient way of tracking changes in both the meta-data and the schema of the hospital variables, and provides information that are necessary for the mapping tasks. The tool acts as a single point of reference and utilizes graphical representations in order to make the data more comprehensive. Moreover, our QCT produces quality reports for both tabular and imaging data, that can be reviewed by a researcher, in order to accept, correct or eliminate records.

8 ACKNOWLEDGMENT

This work was funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 785907 (HBP SGA2).

REFERENCES

[1] K. Adlassnig et al. Requirements regarding quality certification of electronic health records. In *Medical Informatics in a United and Healthy Europe: Proceedings of MIE 2009, the XXII International Congress of the European Federation for Medical Informatics*, volume 150, page 384. IOS Press, 2009.

[2] K. Amunts, C. Ebell, J. Muller, M. Telefont, A. Knoll, and T. Lippert. The human brain project: Creating a european research infrastructure to decode the human brain. *Neuron*, 92(3):574 – 581, 2016.

[3] D. G. Arts, N. F. De Keizer, and G.-J. Scheffer. Defining and improving data quality in medical registries: a literature review, case study, and generic framework. *Journal of the American Medical Informatics Association*, 9(6):600–611, 2002.

[4] D. Boyle and S. Cunningham. Resolving fundamental quality issues in linked datasets for clinical care. *Health Informatics Journal*, 8(2):73–77, 2002.

[5] A. Calimera, E. Macii, and M. Poncino. The human brain project and neuromorphic computing. *Functional neurology*, 28(3):191, 2013.

[6] C. Curdt, D. Hoffmeister, G. Waldhoff, C. Jekel, and G. Bareth. Development of a metadata management system for an interdisciplinary research project. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:7–12, 2012.

[7] E. Falotico, L. Vannucci, A. Ambrosano, U. Albanese, S. Ulbrich, J. C. Vasquez Tieck, G. Hinkel, J. Kaiser, I. Peric, O. Denninger, et al. Connecting artificial brains to robots in a comprehensive simulation framework: the neurorobotics platform. *Frontiers in neurorobotics*, 11:2, 2017.

[8] J. Greenberg, H. C. White, S. Carrier, and R. Scherle. A metadata best practice for a scientific data repository. *Journal of Library Metadata*, 9(3-4):194–212, 2009.

[9] G. Jiang, D. K. Sharma, H. R. Solbrig, C. Tao, C. Weng, and C. G. Chute. Building a semantic web-based metadata repository for facilitating detailed clinical modeling

in cancer genome studies. In *SWAT4LS*. Citeseer, 2014.

[10] H.-E. Källman, E. Halsius, M. Folkesson, Y. Larsson, M. Stenström, and M. Båth. Automated detection of changes in patient exposure in digital projection radiography using exposure index from dicom header metadata. *Acta Oncologica*, 50(6):960–965, 2011.

[11] K. Karozos, I. Spartalis, A. Tsikiridis, D. Trivela, and V. Vassalos. Integrating clinical data from hospital databases. In *First International Workshop on Semantic Web Technologies for Health Data Management*, 2018.

[12] K. A. Kerr, T. Norris, and R. Stockdale. The strategic management of data quality in healthcare. *Health Informatics Journal*, 14(4):259–266, 2008.

[13] P. T. Liu, C. D. Johnson, R. Miranda, M. D. Patel, and C. J. Phillips. A reference standard-based quality assurance program for radiology. *Journal of the American College of Radiology*, 7(1):61–66, 2010.

[14] P. Mildenberger, M. Eichelberg, and E. Martin. Introduction to the dicom standard. *European radiology*, 12(4):920–927, 2002.

[15] L. Orfanidis, P. D. Bamidis, and B. Eaglestone. Data quality issues in electronic health records: an adaptation framework for the greek health system. *Health informatics journal*, 10(1):23–36, 2004.

[16] L. Perimal-Lewis, D. Teubner, P. Hakendorf, and C. Horwood. Application of process mining to assess the data quality of routinely collected time-based performance data sourced from electronic health records by validating process conformance. *Health informatics journal*, 22(4):1017–1029, 2016.

[17] C. Prieto, E. Vano, J. Ten, J. Fernandez, A. Iñiguez, N. Arevalo, A. Litcheva, E. Crespo, A. Floriano, and D. Martinez. Image retake analysis in digital radiography using dicom header information. *Journal of digital imaging*, 22(4):393–399, 2009.

[18] A. Sen. Metadata management: past, present and future. *Decision Support Systems*, 37(1):151–173, 2004.

[19] O. SHABESTARIal and A. Roudsari. Challenges in data quality assurance for electronic health records. *Enabling Health and Healthcare Through ICT: Available, Tailored and Closer*, 183:37, 2013.

[20] L. E. Sweet and H. L. Moulaison. Electronic health records data and metadata: challenges for big data in the united states. *Big data*, 1(4):245–251, 2013.

[21] V. Tassos and V. Vasilis. Data integration in the human brain project. In *International Conference on Data Integration in the Life Sciences*, pages 28–36. Springer, 2015.

Virtual Private Ledgers: Embedding Private Distributed Ledgers over a Public Blockchain by Cryptography

Antonio Arena
antonio.arena@ing.unipi.it
Department of Information Engineering, University of Pisa
Largo Lucio Lazzarino, 1 - 56122
Pisa (Italy)

Pericle Perazzo
pericle.perazzo@iet.unipi.it
Department of Information Engineering, University of Pisa
Largo Lucio Lazzarino, 1 - 56122
Pisa (Italy)

Gianluca Dini
gianluca.dini@iet.unipi.it
Department of Information Engineering, University of Pisa
Largo Lucio Lazzarino, 1 - 56122
Pisa (Italy)

ABSTRACT

Distributed ledgers allow us to replicate databases of records across mutually untrusted parties. The best known example of distributed ledger is perhaps the Bitcoin blockchain, which maintains a consistent history of financial transactions organized as a hashed chain of blocks. Distributed ledgers can be public, i.e., accessible by everyone, or private, i.e., accessible only by a given consortium of parties. In this paper, we explore the technological possibilities of applying Identity-Based Encryption and Attribute-Based Encryption to distributed ledgers. We introduce the novel concept of Virtual Private Ledger. A Virtual Private Ledger is a private distributed ledger embedded in a public cryptocurrency ledger by means of cryptography. A Virtual Private Ledger provides for the same confidentiality and integrity of a private distributed ledger, but without its high operational costs. In particular, nodes that maintain the ledger do not have to be always online to trust the order and the integrity of the records. We analytically show that Virtual Private Ledgers can be implemented over many existing cryptocurrency ledgers like Ethereum, EOS.IO, IOTA, XRP. Different cryptocurrencies lead to different trade-offs between the Virtual Private Ledger max record size, cost, validation time, and max consortium members.

CCS CONCEPTS

• **Networks** → *Peer-to-peer networks*; • **Security and privacy** → *Network security*;

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331083

KEYWORDS

Blockchain, Confidentiality, Distributed Ledger, Feasibility Analysis

ACM Reference format:
Antonio Arena, Pericle Perazzo, and Gianluca Dini. 2019. Virtual Private Ledgers: Embedding Private Distributed Ledgers over a Public Blockchain by Cryptography. In *Proceedings of 23rd International Database Engineering & Applications Symposium, Athens, Greece, June 10–12, 2019 (IDEAS'19),* 9 pages.
https://doi.org/10.1145/3331076.3331083

1 INTRODUCTION

In the last years distributed ledgers have been object of many studies in different IT sectors, such as smart home [13], smart grid [19], healthcare [20], smart city [3], and so on. In general, a distributed ledger is useful when multiple *peers*, possibly having conflicting interests, want to agree on a shared history of records. It allows the peers to trust record order and consistency without trusting any other peer in the network [25]. Distributed ledgers can be public or private. In private distributed ledgers, records are available only to a restricted group of authorized parties, called *consortium*. This is useful when records carry privacy-sensitive or business-critical information.

One problem of employing distributed ledgers on a vast scale is that they require the peers to maintain a complete local copy of the entire ledger [23]. Depending on the specific application, such a ledger can be quite large. For example in Internet of Things applications, data is produced not only by human beings but also by "things". The number of Internet of Things devices is exponentially increasing and it is expected to reach more than 20 billion devices by 2020 [15]. It is also estimated that nearly 850 Zbytes will be generated by all people and things by 2021 [11]. Maintaining copies of such data in all the peers involved in a distributed ledger could be prohibitively costly. Moreover, peers must be constantly online to perform consensus protocol for each block of data to be recorded in the ledger.

The inspiration for this paper comes from Virtual Private Networks (VPN), which allow us to build private networks over public insecure ones by means of cryptography. Analogously, we explore the possibility of implementing a *Virtual Private Ledger* (VPL) by embedding it in an existing cryptocurrency ledger by means of cryptography. Cryptocurrencies often organize their distributed ledger as a *blockchain*, which is a recent and special kind of distributed ledger that maintains a consistent history of records organized as a hashed chain of blocks. Blockchain technology is having a great momentum in the last years due to its application in the widespread Bitcoin cryptocurrency [21]. A VPL provides for the same confidentiality and integrity of a private distributed ledger, but without its high operational costs. In particular, peers do not have to be always online to trust the order and the integrity of the records. They do not have to execute the cryptocurrency consensus algorithm or to maintain a local copy of the whole ledger. The blockchain peers will do it for them as long as they are incentivized by maintaining the consistency of the cryptocurrency transactions. We also give a mechanism by which peers can reach a consensus on the *semantic validity* of the records, and not only their order and integrity. For example, if the private ledger contains GPS traces of the customers of a car insurance, such traces can be checked to be consistent with other measurements, for example those coming from electronic toll collection systems on highways.

The rest of the paper is organized as follows. Section 2 introduces the main technological aspects of distributed ledgers and blockchains. In Section 3 we review some relevant related work. Section 4 introduces the Virtual Private Ledger concept, and our reference threat model. Section 5 analytically investigates the feasibility of embedding a VPL in existing cryptocurrency blockchains. Finally, Section 6 concludes the paper.

2 PRELIMINARIES

A distributed ledger is a replicated database of records shared across a network of multiple sites, geographies or institutions [27]. A distributed ledger is maintained in a distributed fashion, and it does not need for central administration or centralized data storage. All participants have their own identical copy of the ledger. Any changes to the ledger are reflected in all copies in minutes, or in some cases, seconds. In order to reach agreement on the ledger status a *consensus protocol* is needed. Different consensus protocol has been proposed in literature, with different security, scalability and timing properties [29]. Distributed ledgers can be both *public* or *private*. Public distributed ledgers are publicly accessible, so that anyone can download and read the entire record history. Therefore, it is not secure to use a public distributed ledger to carry privacy-sensitive or business-critical information. In contrast, in private distributed ledgers the access to records is restricted to the members of a consortium, which are authorized by a central authority. A consortium member can be a *client* or a peer. The clients simply read and write records on the ledger. The peers are also in charge of executing the consensus protocol. Examples of private distributed ledgers are Hyperledger Indy [1] and Tendermint [9]. Private distributed ledgers typically use variations of the *Practical Byzantine Fault Tolerance* protocol [26] as a consensus protocol. A private distributed ledgers can securely carry personal and critical information, since only the authorized entities can access it. However, their operational cost is generally quite expensive, since all the peers have to maintain the entire record database.

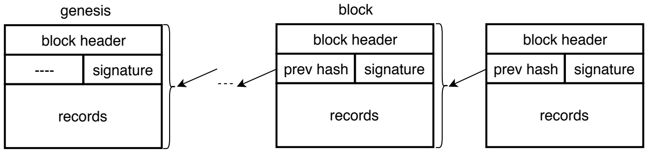


Figure 1: Typical structure of a blockchain

In the last years the distributed ledger technology gained great interest in both academia and industry thanks to introduction of the *blockchain* technology. A blockchain is a distributed, tamper-proof distributed ledger, whose typical structure is shown in Fig. 1. It is a list of ordered blocks, where each block stores a variable-size list of records. Each block is *chained* to the previous one, by including the hash value of it. The blockchain is maintained in a distributed fashion by a set of *peers*, which participate to the consensus protocol. The records not yet included in a block are collected by peers, which gradually fill up a new block that may be different for every peer. Typically, when this new block reaches a predefined maximum size or when a predefined timer expires, a distributed consensus protocol can start. As a result of the consensus protocol, one peer is elected as *temporary central peer*, and it decides the next block to be added to the blockchain. The temporary central peer signs the block and broadcasts it to all peers so that they can verify that the block was built from valid records and possibly append it to their locally maintained blockchain. The *block header* included in this block conveys all the information needed to verify the correctness of the executed consensus protocol. Every blockchain starts with a special block, called *genesis*, which does not reference a previous block and must be known a priori by all the peers. The structure of the blockchain guarantees us that any change on the order or the content of the block records would entirely change the successive blocks of the blockchain. For this reason, changing something in the blockchain would require to run several instances of the consensus protocol again, which is practically unfeasible.

Historically, the first proposed consensus protocol for blockchains has been the *Proof of Work* (PoW) protocol [21], used for example in Bitcoin. The PoW protocol is based on finding a solution of a hard-solving mathematical problem (*puzzle*). The puzzle must be hard to solve, but it must be easy to verify that a solution is correct. A typical puzzle, used in Bitcoin and many other cryptocurrencies, is finding a quantity to include inside the block such that the block's hash is below a predefined target. The peer who first finds a solution automatically becomes the temporary central peer, and it decides the next block to be added to the blockchain. In order to incentivize peers to spend computational resources for solving puzzles, the PoW protocols typically reward somehow the temporary central peer. For example, in Bitcoin the temporary central peer gains a fixed quantity of Bitcoins[1]. The emerging blockchain technology takes a relevant role also in several application scenarios like smart home [13], smart grid [19], healthcare [20], smart city [3], and so on. In general, a blockchain network is useful when multiple entities having conflicting interests want to agree on a shared history of records. It allows nodes to trust record order and integrity without trusting any node in the network [25].

3 RELATED WORK

Dorri et al. [14] proposed a distributed ledger solution for vehicular ecosystems, based on blockchain. Ledger records store the hash values of data generated by in-vehicle sensor, e.g. GPS traces, brakes utilization, traffic information etc. The actual data is instead stored in the vehicle themselves, leveraging a mass storage such as an SD card, and it is retrieved only in case of real need, e.g., car accidents. This allows us to guarantee more privacy over data, and at the same time it reduces the size of the distributed ledger, thus reducing its operational cost. However, data is not available in any moment by nodes. Moreover, it is not possible to verify the semantic validity of data at the moment of a ledger update, so that the ledger could contain invalid data at any moment. For example, if the ledger contains GPS traces for a car insurance application, such traces could be inconsistent with other measurements, for example those coming from electronic toll collection systems on highways. Our proposal reduces the operational cost of the private ledger by embedding it on a public cryptocurrency blockchain. This allows us to guarantee data availability to peers, and to enforce a semantic validity over data.

Zyskind et al. [30] combined a distributed ledger with off-ledger storage to construct a personal data management platform focused on privacy. Data is stored in an off-chain distributed hash-table which is enforced through an access control manager implemented on top of a blockchain. The off-blockchain storage is maintained by a network of trusted nodes or simply by a centralized cloud. In our proposal we do not need an off-blockchain storage since data are encrypted and directly included on a public blockchain. The access control mechanism is not implemented with a blockchain but rather by cryptography.

Hyperledger Indy [1] and Tendermint [9] are private distributed ledger technologies that use variations of the Practical Byzantine Fault Tolerance protocol [26] to reach consensus among peers. Their operational cost is generally quite high, since all the peers have to maintain the entire ledger. Moreover, peers must be constantly online to perform consensus protocol for each block of data to be recorded in the ledger. With our approach, consortium members do not have to be always online to trust the order and the integrity of the records. They do not have to execute the consensus algorithm or to maintain a local copy of the whole ledger. The peers of the underlying public blockchain will do it for them as long as they are incentivized by maintaining the consistency of the cryptocurrency transactions.

4 VIRTUAL PRIVATE LEDGER

A *Virtual Private Ledger* (VPL) is a private distributed ledger embedded inside a public cryptocurrency blockchain by means of cryptography. Broadly speaking, the records of the VPL (*VPL records*) are encrypted and stored inside the optional data of the transactions of the cryptocurrency blockchain. The majority of modern cryptocurrencies (e.g., Bitcoin, Ethereum, etc.) support optional data to be included in every transaction. Note that not all the cryptocurrencies require to perform an actual money transfer in order to store a VPL record, as they allow for transactions involving zero coins. In this way it is possible to include a VPL record in a cryptocurrency transaction without transferring money.

4.1 VPL Architecture

The general architecture of a VPL is shown in Fig. 2. A *VPL member* is an entity that can access the VPL, because it possesses the necessary keys to decrypt and authenticate VPL records. The set of all the VPL members is called the *VPL consortium.* A VPL member can be a *VPL clients* or a *VPL peers.* The VPL clients simply read and write records on the VPL. VPL peers are in charge of executing the *validation protocol.* The validation protocol is a consensus protocol which aims at semantically validate the VPL records. For example, if the ledger contains GPS traces of the customers of a car insurance, such traces can be checked to be consistent with other measurements in the ledger, for example those coming from electronic toll collection systems on highways. The validation protocol can be any kind of consensus protocol, for example a PBFT protocol which reaches a consensus within short times compared to proof-of-work protocols [26] and resists up to 33% malicious VPL peers in the consortium.

[1] At the time of writing, approximately 12.5 BTC.

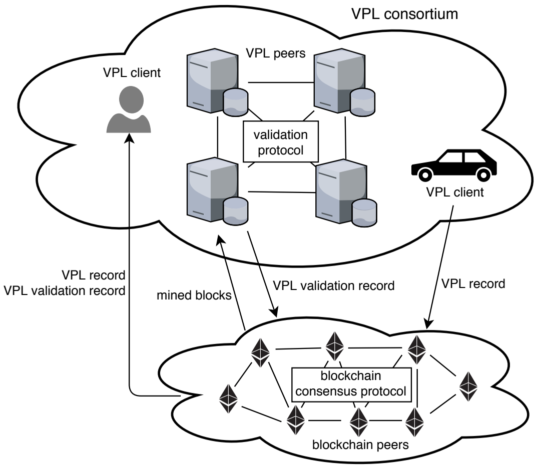


Figure 2: Architecture of a VPL

When a VPL client wants to add some data in the VPL, it encrypts the data thus obtaining in such a way a new VPL record. Then, the VPL client stores the VPL record inside the optional data of a transaction of the public cryptocurrency blockchain. Such a transaction can also involve zero coins. Finally, the VPL client publishes such a transaction in the public cryptocurrency blockchain, and it notices the VPL peers about such new VPL record. If the VPL record needs a semantic validation, then the VPL peers decrypt it and execute the validation protocol. The VPL peers independently check for the VPL record semantic validity and reach a consensus on the outcome. After VPL peers agreed that the VPL record is semantically valid, they publish a *VPL validation record* on the cryptocurrency blockchain, which proves such a validity to the VPL clients. Which specific peer is in charge of publishing the VPL validation record is out of the scope of this paper. For example, the peers can follow a round-robin policy, and reach consensus on which peer must publish the next VPL validation record. Such a consensus can be reached contextually to the execution of the validation protocol.

When a VPL client wants to read some data from the VPL, it simply retrieves the relative VPL record from the public cryptocurrency blockchain, and decrypt it. Depending on the type of encryption employed, access control rules can be enforced on VPL records. For example, VPL clients can read all the records or only a subset of them. If the VPL record underwent a semantic validation, then the VPL client also retrieves the relative VPL validation record from the cryptocurrency blockchain, which proves the validity of the VPL record.

4.2 VPL Records

Fig. 3 shows the format of a VPL record. The VPL record is produced by a VPL client and stored inside the optional data field[2] of a cryptocurrency transaction. The block header and the transaction header depend on the format of the cryptocurrency blockchain. We only require that the transaction header contains a signature of the whole transaction including the optional data, in such a way that the VPL record is signed by the VPL client that produced it. This is assured by all the major cryptocurrencies. The *VPL header* contains an identifier of the VPL consortium to which the record belongs to. It contains also the identifier of the key (or keys) able to decrypt the data. The *encrypted payload* is the actual data encrypted by means of some form of cryptography. We identified two possible forms of suitable cryptography, depending whether the consortium wants to enforce an access control on data. In a VPL without access control, all the clients and peers are authorized to read all the records. The access to such records must be denied only to entities external to the consortium. A VPL without access control can be obtained by encrypting records with Identity-Based Encryption (IBE) [5, 7, 8, 18]. IBE is capable of encrypting a record in such a way that only who has a particular identity can decrypt it afterwards. Such an identity can refer to a single entity as well as to a group of entities. In our case, the VPL records must be encrypted with the identity of the VPL consortium, in such a way that all the VPL members (clients and peers) can decrypt them. On the other hand, in a VPL with access control the clients are authorized to read some of the records, following a fine-grained access control, whereas the peers can access them all. A VPL with access control can be obtained by encrypting records with Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [2, 24]. CP-ABE is capable of encrypting a record in such a way that only who fulfills a particular *access policy* can decrypt it afterwards. Such an access policy is expressed through a Boolean formula computed over the attributes that describe the client. The client is able to decrypt the record only if such a formula evaluates to true. The access policies must be such that the peers can always decrypt any record.

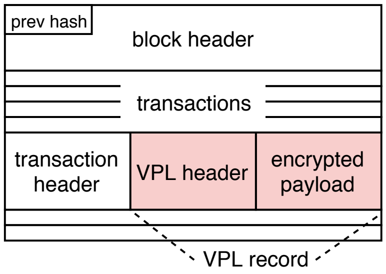


Figure 3: VPL record format

[2] In some cryptocurrencies the optional data field is called *memo field*.

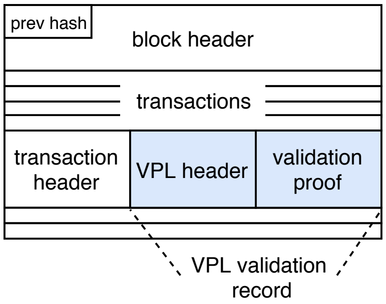


Figure 4: VPL validation record format

4.3 VPL Validation Records

Fig. 4 shows the format of a VPL validation record. The VPL validation record is stored inside the optional data field of a cryptocurrency transaction. Also here, we require that the transaction header contains a signature of the whole transaction including the optional data, in such a way that the VPL validation record is signed by the peer that produced it. The VPL header contains an identifier of the VPL consortium to which the record belongs to. It contains also the identifier of the VPL record that it validates. The *validation proof* contains the proof that the VPL peers reached consensus in declaring the VPL record valid. The validation proof must maintain all the useful information for the VPL clients to correctly determine whether the consensus protocol correctly ended. As we already mentioned above, the consensus protocol can be any kind of consensus protocol, for example a *Practical Byzantine Fault Tolerance* protocol (PBFT), which reaches a consensus within short times and resists up to 33% malicious VPL peers in the consortium. In the case a PBFT protocol is employed, the validation proof must contain the positive outcome of the validation signed by all the VPL peers [10].

4.4 Threat Model

The main adversary against a VPL is a client wanting to store semantically inconsistent data in the VPL, possibly colluding with one or more peers. For example, think about a car insurance application where costumers pay a premium based on how many miles the car traveled. Suppose that costumers store their GPS traces on a VPL that gathers also other independent measurements, for example those coming from electronic toll collection systems on highways. The insurance company computes the premium basing on such GPS traces. A malicious VPL client could try to store GPS traces inconsistent with the records of the electronic toll collection system. In this way, he can pay an insurance premium lower than the real one, computed on fake GPS traces.

In the most simple attack scenario, a malicious VPL client generates a VPL record carrying semantically inconsistent data. However, the VPL peers will reject the semantic validity of this inconsistent VPL record, so they do not generate the VPL validation record. In a more complex attack scenario, the client initially generates a VPL record carrying semantically consistent data, so that the peers will generate a VPL validation record for the VPL record. After the VPL validation record is published on the blockchain, the malicious VPL client tries to modify the encrypted payload of the validated VPL record. However, the malicious client cannot do that since the underlying cryptocurrency blockchain provides for immutability. The VPL client may also collude with one or more VPL peers. In this scenario, the client generates a VPL record carrying semantically consistent data, and the colluding peers try to reach consensus on the validity of this inconsistent VPL record among all the peers, so that a VPL validation record is generated and published on the blockchain. However, the colluding peers cannot do that unless they are more than 33% of the total VPL peers. This is because the PBFT protocol is employed to reach consensus on the semantic validity for a VPL record.

5 FEASIBILITY ANALYSIS

In this section we study the feasibility of implementing a Virtual Private Ledger over a public cryptocurrency blockchain. To this aim, we considered the Boneh-Franklin (BF) encryption scheme [6] and the Bethencourt-Sahai-Waters (BSW) encryption scheme [2], which represent classic schemes respectively for IBE and CP-ABE. We remind that IBE is capable of encrypting a VPL record in such a way that only who has a particular identity can decrypt it afterwards. In our case, the VPL records must be encrypted with the identity of the VPL consortium, in such a way that all the VPL members can decrypt them. On the other hand, CP-ABE is capable of encrypting a VPL record in such a way that only who fulfills a particular access policy can decrypt it afterwards, thus granting a fine-grained access control with which VPL member can decrypt a VPL record.

We considered the minimal set of fields that must be included in each VPL record and each VPL validation record. A VPL record must always include a *VPL record identifier*, which is needed to relate a VPL validation record with a VPL record, and a *VPL consortium identifier*, which is an identifier for distinguishing different Virtual Private Ledgers on the same cryptocurrency blockchain. We assume VPL records identifiers are on 16 bytes and VPL consortium identities on 8 bytes. We also took into consideration the *encryption overhead*, that is the size difference between a plaintext and the corresponding ciphertext, which depends on the employed encryption scheme. Assuming a security level of 80 bits, the BF encryption scheme adds an encryption overhead of 84 bytes [6], whereas the BSW encryption scheme adds an encryption overhead which depends on the access policy with

Table 1: VPL record and VPL validation records fields dimensions

Field	Size
VPL record identifier	16 bytes
VPL consortium identifier	8 bytes
BF encryption scheme overhead[3]	84 bytes
BSW encryption scheme overhead[3] [4]	832 bytes
5-attribute policy representation	30 bytes
VPL validation record identifier	16 bytes
Semantic validation outcome	4 bytes
VPL peer identifier	4 bytes
ECDSA signature[3]	40 bytes

which the client encrypts the VPL record. Supposing a security level of 80 bits and access policies of 5 attributes, the BSW encryption scheme adds an encryption overhead of 832 bytes. If a CP-ABE scheme is employed, we need to embed in the VPL record also a representation of the access policy that a VPL member needs to fulfill for decrypting the VPL record. An access policy is a Boolean formula specified by the client that encrypts the VPL record. We assume access policies of 5 attributes are represented with strings of 30 bytes. To sum up, the minimum VPL record size (not counting the data payload which depends on the specific application) is 108 bytes if the VPL does not employ access control mechanisms, and 886 bytes if the VPL employs an access control mechanism with access policy of 5 attributes.

A VPL validation record must always include a *VPL validation record identifier* and the identifier of the VPL record that it is validating. We assume VPL validation records identifiers are on 16 bytes. The VPL validation record must also carry the *semantic validation outcome*, i.e., whether the data carried in the VPL record were semantically valid or not, and the proof that the VPL peers reached consensus in the declaring the VPL record as valid or not. Assuming a VPL consortium employs a PBFT protocol among N peers, this proof must include N signatures of the semantic validation outcome [10], along with the corresponding peer identities, represented by *VPL peer identifiers*. We assume the semantic validation outcomes are on 4 bytes, and the peer identifiers on 4 bytes. We assume to use the Elliptic Curve Digital Signature Algorithm (ECDSA) with a security level of 80 bits as signature algorithm, which results in a signatures of 40 bytes [17]. To sum up, the VPL validation record size is $36 + 44 \times N$ bytes. Table 1 summarizes the sizes for all the fields of the VPL records and of the VPL validation records.

Table 2: Available space in optional data fields with different cryptocurrencies

Cryptocurrency	Optional data max size
Bitcoin	83 bytes [5]
EOS.IO	256 bytes [6]
Ethereum	98,225 bytes [7]
IOTA	1300 bytes [8]
Stellar	28 bytes [9]
XRP	1024 bytes [10]

Table 2 summarizes the maximum available space that can be used for embedding a VPL record and a VPL validation record in the optional data field of a transaction for six different cryptocurrencies, namely, Bitcoin, EOS.IO, Ethereum, IOTA, Stellar and XRP. We assume to embed VPL records and VPL validation records into a single cryptocurrency blockchain, as fragmenting a single record in multiple cryptocurrency transactions would results in high costs and high delays in publishing them in a block. By considering the VPL record and VPL validation record dimensions described above, it is impossible to embed them into a single Bitcoin or Stellar transaction, thus making these cryptocurrencies unable to host a Virtual Private Ledger. The EOS.IO blockchain can be chosen as underlying blockchain only if the VPL does not employ an access control mechanism for VPL records. Finally, the Ethereum, IOTA and XRP blockchains offer enough available space in a single transaction to host a Virtual Private Ledger.

Finally, in Table 3 we compare the maximum payload dimension for every feasible blockchain, the minimum cost for generating a cryptocurrency transaction[11], the maximum number of VPL peers in a VPL consortium and the average *transaction confirmation time*. The transaction confirmation time is defined as the time elapsed between the moment a cryptocurrency transaction is submitted to the blockchain and the time it is recorded into a reached consensus on a block. In other words, it represents the total time a VPL client has to wait until a transaction gets collected and included a block in the blockchain. As we already said, the EOS.IO cryptocurrency is feasible only if a VPL consortium does not need access control for VPL records. As its maximum optional data size is quite low, the maximum number of VPL peers cannot

[3] Assuming a security level of 80 bits.

[4] Assuming a fixed size of the attributes set equal to 5.

[5] http://bit.ly/bitcoin_transaction_size

[6] http://bit.ly/EOS_memo_size

[7] http://bit.ly/ethereum_transaction_size

[8] http://bit.ly/iota_transaction_size

[9] http://bit.ly/stellar_memo_size

[10] http://bit.ly/xrp_memo_size

[11] Exchange values computed at April 11, 2019

Table 3: Comparison of different feasible cryptocurrency blockchains

Cryptocurrency	Max payload (no access control)	Max payload (with access control)	Minimum transaction cost	Transaction confirmation time	Max peers N
EOS.IO	148 bytes	(unfeasible)	0 $	0.5 s	5
IOTA	1192 bytes	414 bytes	0 $	[2, 10] m	28
XRP	916 bytes	138 bytes	0.00000337 $	3.6 s	22
Ethereum Max optional data size = 1000 bytes	892 bytes	114 bytes	0.059$	30 s	21
Ethereum Max optional data size = 5000 bytes	4892 bytes	4114 bytes	0.24$	30 s	112
Ethereum Max optional data size = 10000 bytes	9892 bytes	9114 bytes	0.46$	30 s	226

exceed 5 peers. However, the average transaction confirmation time is very low, as in EOS.IO a block is generated exactly every 0.5 seconds [4], and it is not required to pay for generating transactions. The IOTA cryptocurrency offers instead a quite large optional data field, leaving more than 400 bytes for the payload in case a VPL consortium needs access control for VPL records. Moreover, we can have up to 28 VPL peers in a consortium and it is possible to generate transaction without transferring money. However, the IOTA transaction confirmation time is quite variable as it depends on the global transaction generation rate. In the worst case, a transaction may be confirmed even after 10 minutes [12]. Furthermore, with IOTA the VPL client must solve a PoW puzzle for every VPL record it generates, which is practically unfeasible if VPL client is implemented on a battery-powered device [16]. In case a VPL consortium need access control policies for VPL records, the XRP cryptocurrency is suitable only if a VPL record carries a small amount of data, i.e., max 138 bytes. In XRP a VPL member has to pay a very small amount of money for generating a transaction. Moreover, the XRP cryptocurrency requires that a node must own at least 20 XRP [12] for generating a transaction, otherwise the transaction would never be included in a block [22]. For this reason, a VPL member has to pay for a very small amount of dollars for generating VPL records or VPL validation records. It is important to notice that XRP offers a good transaction confirmation time, which is generally below 4 seconds.

The Ethereum cryptocurrency needs to be analyzed in multiple scenarios. In particular, the cost for an Ethereum transaction depends on its size [28]. For this reason, we analyzed three different Ethereum optional data sizes, i.e., 1000 bytes, 5000 bytes and 10000 bytes. For every size, we computed the maximum number of VPL peers and the cost of a single transaction. The Ethereum cryptocurrency offers a very large optional data field and a moderate transaction confirmation time. However, the cost for a transaction becomes prohibitive for large VPL records, up to 0.46$ for a single VPL record. If a VPL client generates multiple VPL records in a day, the cumulative cost for a VPL client becomes very high. Moreover, the VPL peers may not be encouraged to generate VPL validation records, if they have to pay for publishing them. In this situation, a VPL client may be forced by the VPL consortium to pay an extra fee to the VPL peers to cover the cost of a VPL validation record, thus making Ethereum quite expensive.

In conclusion, we consider the IOTA cryptocurrency the best choice for hosting a Virtual Private Ledger, if a VPL consortium does not have timing constraints and if clients do not have hardware constraints for performing PoW puzzles. With IOTA, we can also have a moderate number of VPL peers in the consortium, and mostly important it is not needed to pay for generating IOTA transactions. If we need a lower transaction confirmation time instead, the XRP cryptocurrency may be the best choice, especially if the VPL consortium does not need access control policies for records. We finally consider the Ethereum cryptocurrency as the optimal one only if members need to generate very large VPL records and a large number of VPL peers is needed, as the cost becomes very high.

6 CONCLUSIONS

The inspiration for this paper came from Virtual Private Network (VPN) technology, which allows us to build a private

[12] At the time of writing, 20 XRP equals to 6.75$

network over a public insecure one by means of cryptography. Analogously, in this paper we explored the possibility of implementing a Virtual Private Ledger (VPL) by embedding it on an existing cryptocurrency blockchain by means of cryptography. A blockchain is a recent and special kind of distributed ledger, which is having a great momentum in the last years due to its application in the widespread Bitcoin cryptocurrency. A VPL provides for the same confidentiality and integrity of a private distributed ledger, but without its high operational costs. We presented a general architecture for a VPL, by giving a guideline for implementing it on top of a generic cryptocurrency blockchain. In this paper we explored the technological possibilities of applying Identity-Based Encryption (IBE) and Cypertext-Policy Attribute-Based Encryption (CP-ABE). Finally, we discussed the possibility of implementing a VPL on top of different existing cryptocurrency blockchains, such as Bitcoin, EOS.IO, Ethereum, IOTA, Stellar and XRP. We found out that the Bitcoin and Stellar cryptocurrencies are not suitable as they do not offer enough space for embedding encrypted data in a single transaction. The IOTA, EOS.IO and XRP cryptocurrencies are instead the best choices for implementing a VPL on top of them, as they offer enough free space for encrypted data and they do not need to pay for generating transaction.

ACKNOWLEDGEMENTS

This work was partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence) and by the University of Pisa in the framework of the project PRA_2018_81 "Wearable sensor systems: personalized analysis and data security in healthcare".

REFERENCES

[1] [n. d.]. Hyperledger Indy. ([n. d.]). https://www.hyperledger.org/projects/hyperledger-indy

[2] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*. IEEE, 321–334.

[3] Kamanashis Biswas and Vallipuram Muthukkumarasamy. 2016. Securing smart cities using blockchain technology. In *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on*. IEEE, 1392–1393.

[4] Block.one. [n. d.]. EOS Documentation. ([n. d.]). https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md#consensus-algorithm-bft-dpos

[5] Dan Boneh and Matt Franklin. 2001. Identity-based encryption from the Weil pairing. In *Annual international cryptology conference*. Springer, 213–229.

[6] Xavier Boyen. 2008. A tapestry of identity-based encryption: practical frameworks compared. *IJACT* 1, 1 (2008), 3–21.

[7] Francesco Buccafurri, Gianluca Lax, Lorenzo Musarella, and Antonia Russo. 2019. Ethereum Transactions and Smart Contracts among Secure Identities. In *Proceedings of the Second Distributed Ledger Technology Workshop, DLT@ITASEC 2019, Pisa, Italy, February 12, 2019.* 5–16.

[8] Francesco Buccafurri, Gianluca Lax, Antonia Russo, and Guillaume Zunino. 2018. Integrating Digital Identity and Blockchain. In *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*. Springer International Publishing, Cham, 568–585.

[9] Ethan Buchman. 2016. *Tendermint: Byzantine fault tolerance in the age of blockchains*. Ph.D. Dissertation.

[10] Miguel Castro, Barbara Liskov, et al. 1999. Practical Byzantine fault tolerance. In *OSDI*, Vol. 99. 173–186.

[11] Cisco. [n. d.]. Cisco Global Cloud Index: Forecast and Methodology, 2016âĂŞ2021 White Paper. http://bit.ly/2DHLOYO. ([n. d.]). [Online].

[12] Cyclux. [n. d.]. TangleMonitor. ([n. d.]). https://tanglemonitor.com

[13] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. 2017. Blockchain for IoT security and privacy: The case study of a smart home. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 618–623.

[14] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak. 2017. BlockChain: A Distributed Solution to Automotive Security and Privacy. *IEEE Communications Magazine* 55, 12 (DECEMBER 2017), 119–125. https://doi.org/10.1109/MCOM.2017.1700879

[15] Nathan Eddy. 2015. Gartner: 21 Billion IoT devices to invade by 2020. *InformationWeek, Nov* 10 (2015).

[16] Atis Elsts, Efstathios Mitskas, and George Oikonomou. 2018. Distributed ledger technology and the internet of things: A feasibility study. In *Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems*. ACM, 7–12.

[17] Don Johnson, Alfred Menezes, and Scott Vanstone. 2001. The elliptic curve digital signature algorithm (ECDSA). *International journal of information security* 1, 1 (2001), 36–63.

[18] Allison Lewko and Brent Waters. 2011. Decentralizing attribute-based encryption. In *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 568–588.

[19] Esther Mengelkamp, Benedikt Notheisen, Carolin Beer, David Dauer, and Christof Weinhardt. 2018. A blockchain-based smart grid: towards sustainable local energy markets. *Computer Science-Research and Development* 33, 1-2 (2018), 207–214.

[20] Matthias Mettler. 2016. Blockchain technology in healthcare: The revolution starts here. In *e-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on*. IEEE, 1–3.

[21] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[22] XRP Developers Portal. [n. d.]. XRP: Reserves. ([n. d.]). https://developers.ripple.com/reserves.html

[23] Bitcoin Project. [n. d.]. Bitcoin Core Requirements and Warnings. ([n. d.]). https://bitcoin.org/en/bitcoin-core/features/requirements

[24] Marco Rasori, Pericle Perazzo, and Gianluca Dini. 2018. ABE-Cities: An Attribute-Based Encryption System for Smart Cities. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 65–72.

[25] Sarah Underwood. 2016. Blockchain Beyond Bitcoin. *Commun. ACM* 59, 11 (Oct. 2016), 15–17. https://doi.org/10.1145/2994581

[26] Marko Vukolić. 2015. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *International Workshop on Open Problems in Network Security*. Springer, 112–125.

[27] MGCSA Walport et al. 2016. Distributed ledger technology: Beyond blockchain. *UK Government Office for Science* 1 (2016).

[28] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151 (2014), 1–32.

[29] Yang Xiao, Ning Zhang, Jin Li, Wenjing Lou, and Y Thomas Hou. 2019. Distributed Consensus Protocols and Algorithms. *Blockchain for Distributed Systems Security* (2019), 25.

[30] Guy Zyskind, Oz Nathan, et al. 2015. Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 180–184.

On Discovering Relevant Features for Tongue Colored Image Analysis

Eugenio Vocaturo†
DIMES - Department of Computer Science, Modelling, Electronic and System Engineering
University of Calabria
Rende, Italy
e.vocaturo@dimes.unical.it

Ester Zumpano
DIMES - Department of Computer Science, Modelling, Electronic and System Engineering
University of Calabria
Rende, Italy
e.zumpano@dimes.unical.it

Pierangelo Veltri
DSMC- Bioinformatics Laboratory
Surgical and Medical Science Department
University Magna Graecia
Catanzaro, Italy
veltri@unicz.it

ABSTRACT

Artificial Intelligent Systems are increasingly used to support early diagnosis of multiple relevant diseases. The spread of these systems is boosted by the application of machine learning techniques on datasets (also in the form of videos and images) obtained from different information sources. A key role is played by artificial vision systems that are in charge of reasoning on data acquired from different devices, including smartphones. The facility to disseminate and share information let to the globalization of medical protocols previously used just in some world's areas. This is the case of tongue inspection, widely used in Traditional Chinese Medicine (TCM) to perform a diagnosis, which allows physicians to obtain useful indications on the state of internal organs by observing the color and the consistency of patient's tongue. The current interest in tongue's image analysis is also motivated by the possibility of performing a first self-analysis on a possible disease suggesting further medical investigation. The paper is a non-exhaustive overview of the features most frequently used in artificial vision systems contextualized to tongue analysis. It highlights shortcomings in some of the existing studies and provides insights for future research. Our work aims to provide a unifying view that can support the researchers working on Tongue Colored Image Analysis.

CCS CONCEPTS

• Tongue Colored Image Classification • Geometric Features • Texture Features • Color Features

KEYWORDS

Machine Learning, Tongue Image Classification, Features Descriptor

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331124

ACM Reference format:

Eugenio Vocaturo, Ester Zumpano and Pierangelo Veltri. 2019. On Discovering Relevant Features for Tongue Colored Image Analysis. In *IDEAS 2019: 23rd International Database Engineering & Applications Symposium, June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3331076.3331124.

1. Introduction

Until recently, family doctor had a great ability to observe and evaluate every detail of the body, to grasp changes in the normal physiology and signs of disease in progress. Today, however, much weight is given to the results of laboratory analysis, giving little importance to what emerges from the visit: the risk is that therapy is guided by the values of laboratory tests and not by results of patient's medical visit. As for the tongue, notice that in the recent past its analysis was a task systematically performed by the doctor at each visit. Today this is done much less or not at all. Too bad, because tongue analysis can really provide many interesting information and suggest the first treatment. In Traditional Chinese Medicine tongue analysis is greatly taken into account and is used as an extraordinary diagnostic method as it has been proved that correlations exist between the appearance of the tongue and the patient's health status. According to different branches of Oriental Medicine [26] different areas of tongue reflect the functioning of internal organs (Fig. 1).

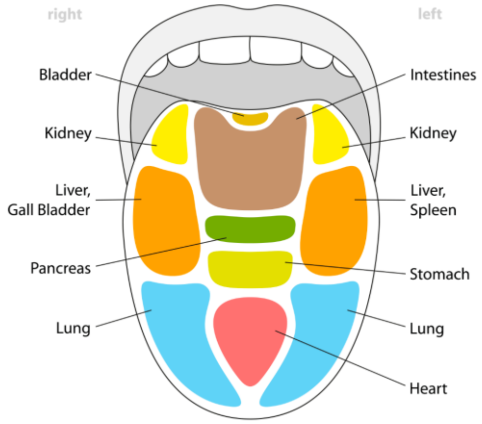


Figure 1: Tongue Reflexology Chart

The tongue is divided transversally into three parts, each corresponding to some organs: kidneys, adrenal and intestines in the proximal third, spleen, liver, pancreas, stomach in the middle third and lungs and heart in the distal third.

The median longitudinal line that divides the tongue into two represents the spinal column. The central part that covers the three sectors and the three main digestive organs, stomach, small intestine and colon, is related to the digestion phases.

Since ancient times, in-depth studies have been carried out in China, the first treatises on this topic go back to the Shang Dynasty (1600-1000 BC). Starting from the 1980s, always in China, systematic studies published in medical journals, have been conducted, on the correlation between the appearance of the tongue and some kind of cancer. In particular, the TCM Association, the Chinese Oncology Association and the TCM Diagnosis Association conducted a national project involving 12.448 cancer patients, 1.628 patients suffering from other diseases and 5.578 healthy people. The results showed that the tongue in patients affected by cancer highlights signs (such as color, shape, mucus layer, etc.) statistically significant in pathognomonic terms compared to that of healthy patients.

Therefore, it is quite evident that physicians can obtain many insights from a correct analysis of the tongue. Often the tongue like the wrist can tells in advance the affections that could eventually emerge. Obviously, to achieve a correct diagnosis tongue signs have to be integrated with additional patient's health information.

2. How is the tongue made?

The tongue is an organ of the human body that occupies most of the oral cavity; it is composed of various anatomical structures: mucous membranes, lingual papillae (also known as taste buds) and various muscles. It constitutes the anterior wall of the oropharynx. Its dorsal surface constituted by the lingual mucosa is convex in every direction and can be divided in two parts, different both for appearance and in their embryological origin, called body and root of the tongue, or oral portion and pharyngeal portion. They are divided by an inverted V-shaped groove called the terminal groove, the apex of which constitutes a small cavity called the blind bottom. It is connected posteriorly to a small bone called a hyoid and anteriorly to a small and thin filament called a frenulum or a thread. The tongue is endowed with taste buds, and is, in fact, the main organ of taste. It performs the function of kneading food with saliva and pushing it under the teeth to be crushed, and then pushed down the esophagus. The body of the tongue constitutes 2/3 of its volume, is longitudinally divided by the median groove, which originates posteriorly at the apex of the tongue and terminates anteriorly to the terminal groove, near the blind hole. With the mouth closed, the lower surface of the tongue body is in contact with the floor of the mouth, the apex with the upper incisors, the lateral margins with the gingival arches and the upper surface with the hard palate and the soft palate. The dorsal surface is covered by a transparent whitish patina consisting of the precipitate on the palate coming from the stomach exhalations through the esophagus. The color, thickness, consistency and ease in removing any present patina give rise to indications on the state of the *digestive function*. On the upper surface of the tongue, anterior to the palatoglossal arch and posterior to the terminal furrow there is an area in which there are 4-6 mucous folds that constitute the residues of foliate papillae, present and functional in many animals, but not in humans. The taste buds are distinguished in:

•***threadlike,*** which appear in the form of a diffused and tiny punctuation, are spread over the entire dorsal surface of the body of the tongue, in particular at the apex,

•***fungiform***, small, raised and rounded, less numerous than the filiform and also distributed over the entire surface,

•***circumvallate,*** more detected and rounded than the others, arranged only along the terminal groove.

The mucosa of the lower surface is red and has a slimy consistency. Two mucosal growths, called fimbriate folds, originate posteriorly and laterally at the base of the tongue and are directed antero-medially defining a triangular area. Medially to these, superficially and following their course, the two deep lingual veins branch off. The lingual frenulum instead connects the lower surface of the tongue with the floor of the mouth. Laterally, at its base, the two sublingual papillae are placed from where the ducts of the submandibular glands open up, through an orifice. On the contrary, the orifices of the sublingual glands are numerous and placed post-laterally with respect to those of the submandibular.

The root of the tongue includes the posterior part of the tongue, i.e. the one comprised anteriorly between the palatoglossal arch and posteriorly between the palatopharyngeal arch. Its surface has vaguely rounded reliefs that make up the protrusion of lymphatic nodules immersed in the lamina of the lingual mucosa; the set of lymphatic nodules constitutes the lingual tonsil.

On the apex of each nodule, the ducts of tubulo-acinar glands open up. Back and side to the lingual tonsil there are the two palatine tonsils, about 1 cm long, housed in spaces between the palatoglossal arch and the palatopharyngeal arch, called palatine pits. Back and inferior to the lingual tonsil there is a plica of elastic cartilage, the epiglottis, that has two lateral glossoepiglottic fold and a median one.

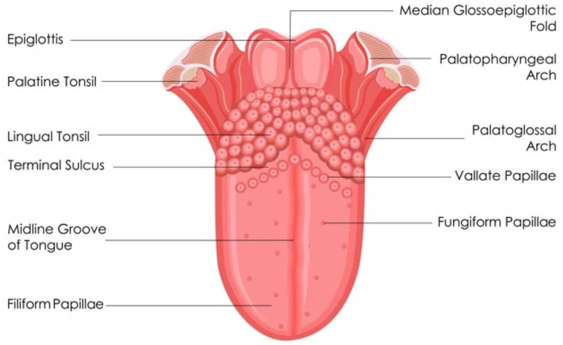


Figure 2: Anatomy of the tongue

3. A road map on tongue image classification

The diagnosis of the tongue is one of the most successful lines of research in complementary medicine together with those of palpation of the wrist and abdomen [1]. The coloring of the tongue, the structure and the geometrical conformation, are placed in correlation with the pathologies of some diseases. The objective is to define protocols in clinical practice and to improve the contribution of computerized systems for the analysis of tongue images. In traditional oriental medicine, tongue color is a discriminating element for diagnosing diseases due to physical and mental disorders such as blood congestion, water imbalance and psychological problems [2].

A group of researchers used a tongue color gamut descriptor, providing the use of SVM for the classification phase [3, 4]. These researches confirmed that the color range of the tongue is very narrow and varies in shades of red. To the naked eye, the colors on different regions of the tongue could appear almost similar, then not permitting a correct diagnosis. Current research in the computerized tongue image analysis system uses machine learning techniques to achieve a superior accuracy rate and a shorter execution time. Therefore, choosing the most effective features to carry out the classification phase therefore becomes necessary. The adoption of too many features implies a complex descriptive mapping in the classifier [5, 6].

TCM has a millennial experience, and boasts significant healing benefits with little side effects. Modern medicine, on the other hand, is focused on the cause-effect relationship: sometimes the side effects are important both in the medium and in the long term. Compared to modern medicine, some practices of TCM are potentially applicable in health, and rehabilitative care protocols. There are four fundamental diagnostic methods in TCM: inspection, smell, interrogation and palpation [7].

The diagnosis of the tongue is one of the most current topics in the field of medicine both for the diagnostic potential of analysis on digital images and for the simplicity with which such images can be obtained. Images of the tongue are increasingly used in clinical work: the recovery of images and their computer management has become a difficult topic. Traditional management of tongue images foresees a manual labeling of the information regarding images and a research phase based on previously obtained information. However, in so doing the needs related to the efficient recovery of images in large-scale datasets cannot be met. On the other hand, traditional tongue diagnosis depends on doctors' experience and it is likely that different doctors will produce different diagnoses for the same patient. Recently, automatic image processing technologies have been applied to support tongue diagnosis in traditional Chinese medicine. These methods through the definition and use of geometry, texture and color features allow the inspection of tongue image [8, 9]. More specifically, in [8, 10] tongue images are managed in different color spaces with different metrics and a method for color recognition on tongue images, based on region partition is proposed.

Li and Yuen, in [8], have addressed the problem of matching color images in medical diagnosis by presenting an ordered metric in the coordinate space. Wang et al., in [10], proposed a new tongue color calibration scheme and used a model based on a vector gradient snake (GVF) that integrates the color information to extract the body of the tongue. The papers [11, 12] share applications that require the use of color, texture or shape features to classify tongue images. In particular, Chiu [11] built a computerized tongue examination system (CTES) based on a chromatic and structural algorithm that identifies the colors of the tongue and the thickness of its coating. Guo [12] proposed a new operator for the color structure, the local binary pattern of the primary difference signal. The corresponding performances are evaluated on color, grayscale and color structure and fusion of color and texture features. Li and Liu developed a hyperspectral brush tongue imager and discussed the method of calibration of the spectral response [13]. This new approach to color analysis outperforms the traditional method, allowing significant areas of tongue substances and coatings to be reached. Each of these methods has its fair share of success, but also limitations regarding demands of precision and robustness. It is necessary to investigate on features to be used to obtain a better analysis of tongue images.

4. Features for Tongue Diagnosis System

Tongue inspection allows an immediate diagnosis of some pathologies, and for this reason, it is widespread in clinical medicine. However, the potential of this examination is limited in traditional diagnosis. The first reason is that the tongue is visually observed by the human eye instead of being analyzed by a quantitative digital instrument. Secondly, the evaluation process is subjective, being linked to the experience and knowledge of the doctor making the diagnosis. Obviously, subjectivism is overcome with the introduction of computerized systems that allow tongue analysis to become objective and repeatable by performing tongue image analysis.

Computerized tongue diagnosis system support the storage and transmission of digital data, as well as image analysis. The typical schema of such a system is reported in Fig. 3 and consists of four different phases: image acquisition, preprocessing, features analysis and classification.

The acquisition of digital images constitutes the first phase in the computer vision system, including many techniques used to acquire the image to be analyzed. Each of those techniques presents advantages and disadvantages that make it more or less suitable depending on the case.

The most used methods for automated non-invasive diagnosis include photography, confocal scanning, laser microscopy (CSLM), ultrasound, magnetic resonance imaging (MRI), optical coherence tomography (OCT), multispectral imaging, computed tomography (CT), positron emission tomography (PET), multi-frequency electrical impedance and Raman spectra.

Preprocessing is the stage of detection used to improve the quality of images, trough color correction and removal of irrelevant noises that may cause inaccuracies in classification [42].

A first goal is to separate, for examples by edge detection techniques, tongue from background.

The quickest way to remove defects related to image acquisition is to use filters such as, medium filters, median filter or Gaussian filters [18]. These filters can be applied directly on grayscale images, and are applied to each channel on color images (marginal filtering).

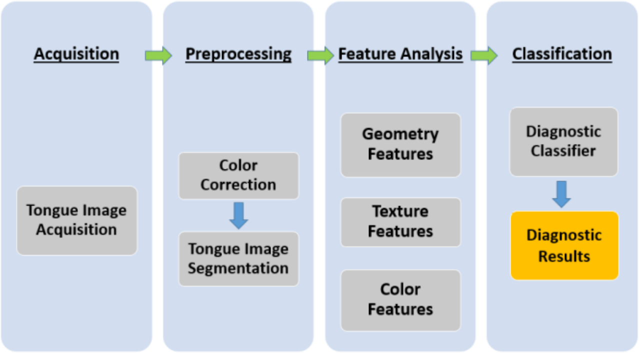


Figure 3: Typical Computerized tongue diagnosis system

In this paper we investigate about the selection of relevant features, useful for implementing medical-type applications for tongue analysis. These features, that will be detailed in next sections, are reported in Fig. 3 and are of three main types, namely geometric, texture and color. Much work has been done to accurately and effectively extract those features whose adoption is fundamental for a sound usability of the systems. In the following sections, we will report, distinguished by type, the most relevant features used for the implementation of medical applications in tongue analysis.

4.1 Geometry Features

Various proposals for computerized tongue diagnosis systems focus on the use of color and texture features of the images [14, 15]. Despite the fact that oriental medicines use the form of tongue as an element of evaluation to discriminate the presence of possible diseases [16, 17, 18], there is poor literature on the geometric features functional to a computerized tongue diagnosis system.
In this section, we will focus on the main geometric features useful to implement models that determine whether a given tongue belongs to a particular type of form. The most commonly referred tongue types are Triangular, Round, Ellipse, Square ad Rectangular (Fig. 4).

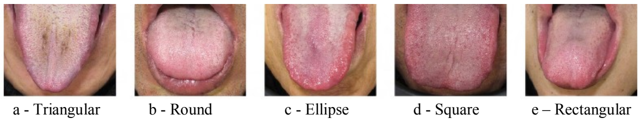


Figure 4: Examples of tongue shape

Using appropriately the geometric features it is possible to discern the shape of the tongue. Experimental results have shown that better classification accuracy can be obtained on tongue images previously separated by shape [19, 20]. In more details, images are first preprocessed to obtain the binary mask that separates the body of the tongue from the outline [15]; subsequently geometric patterns are used to derive the shape of the tongue. The separation of region of interest is a common step that is also adopted in other medical fields [43].
Below, we report the geometric features useful for obtaining the type of tongue shape.

Width. The width (w) feature (Fig.5) is the horizontal distance along the x-axis from a tongue's furthest right edge point ($x_{\max}$) to its furthest left edge point ($x_{\min}$):

$$w = x_{\max} - x_{min} \tag{1}$$

Length. The length (l) feature (Fig. 5) is the vertical distance along the y-axis from a tongue's furthest bottom edge ($y_{\max}$) point to its furthest top edge point ($y_{\min}$):

$$l = y_{\max} - y_{min} \tag{2}$$

Length-Width Ratio. The length-width ratio (lw) is defined as the ratio of a tongue's length to its width:

$$lw = \frac{l}{w} \tag{3}$$

Smaller Half Distance. Smaller half distance (z) is defined as the shorter half distance of l or w (Fig. 5):

$$z = \frac{\min(l,w)}{2} \tag{4}$$

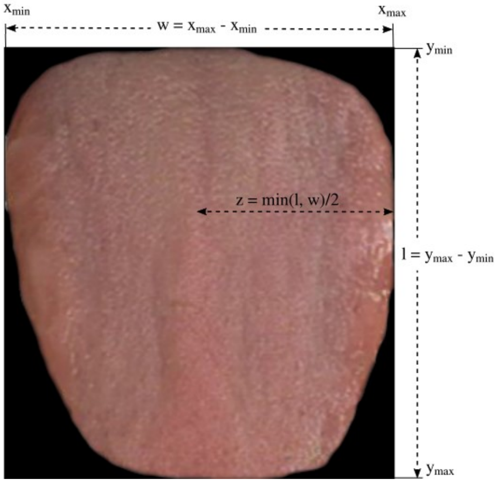


Figure 5: Illustration of features (1), (2), and (4)

Center Distance. The center distance d_c (Fig. 6(a)) is the distance from w's y-axis center point to the center point of y_{d_c}:

$$\begin{cases} c_d = \dfrac{\max(y_{x_{max}})+\max(y_{x_{min}})}{2} - y_{d_c} \\ y_{d_c} = \dfrac{y_{\max}+ y_{min}}{2} \end{cases} \tag{5}$$

Center Distance Ratio. Center distance ratio d_{cr} is ratio of d_c to l:

$$c_{dr} = \frac{c_d}{l} \tag{6}$$

Area. The area (a) refers to the surface measurement of the pixels belonging to the considered tongue's image.

Circle, Square and Triangle Areas. Circle area (c_a) and Square area (s_a) within the tongue are defined with the same radius of the smallest half-distance z (4): see Fig. 6 (b) and Fig. 6 (c).
Triangle area (t_a) is the area of a triangle within the tongue (Fig. 6(d)). Considering $x_{\max}$ and $x_{\min}$ as the right and the left points, and $y_{\max}$ as the bottom point of the triangle, we obtain:

$$c_a = \pi r^2 = \pi z = \pi \left(\frac{\min(l,w)}{2}\right)^2 \tag{7}$$

$$s_a = 4z^2 = 4\left(\frac{\min(l,w)}{2}\right)^2 \tag{8}$$

$$t_a = \frac{bh}{2} \tag{9}$$

where *b* and *h* are respectively the base and the high of a rectangle triangle.

Circle, Square and Triangle Area Ratio. Circle, Square and Triangle area ratio (c_{ar}, s_{ar}, t_{ar}) are the ratio of the considered area to a:

$$c_{ar} = \frac{c_a}{a} \tag{10}$$

$$s_{ar} = \frac{s_a}{a} \tag{11}$$

$$t_{ar} = \frac{t_a}{a} \tag{12}$$

By, appropriately, using the above described features, various approaches for image analysis allowing the effective identification of tongue form have been proposed [21, 22].

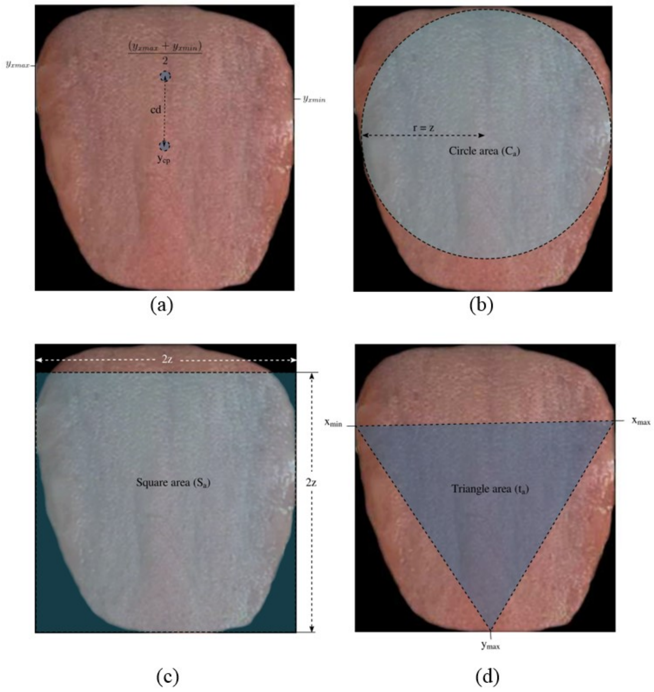


Figure 6: (a) feature (5), (b) feature (7), (c) feature (8) and (d) feature (9)

4.2 Texture Features

A texture can be defined as a two-dimensional image that represents information about the structure of a particular surface. Textures are variations in intensity or color, originating from the roughness of the surfaces of objects hit by a light source. For our purposes, the adoption of appropriate texture features, within an image coding process, facilitates the classification of the analyzed tissues as healthy or pathological. Textures can be divided into two main categories, statistical or related to spatial frequency [24].

Statistical approaches typically consider a discrete set of pixels around a fixed one in order to evaluate the properties that relate the single pixel with its neighborhood. Statistical approaches evaluate various properties and are suitable when texture primitive sizes are dimensionally comparable with the pixel sizes. These properties include among others Fourier transforms, convolution filters, co-occurrence matrix, spatial autocorrelation and fractals. Through these properties, it is possible to identify different textures with respect to specific pre-selected parameters, based on distributions of the gray levels or color channel of the pixels, composing the image.

Spatial frequency approaches evaluate the image in the domain of its frequencies to recognize the patterns [23, 24].

In fact, the tone and structural relations between the primitives of an image allow to identify the plots within the same image. The tone depends on the intensity of the pixels, in terms of gray values or color channel values, in primitives, whereas the structure is related to the spatial configuration among primitives. Therefore, a given pixel can be characterized by its tone and position properties. Primitive texture means a contiguous set of pixels characterized by a certain tone and can be described by means intensity, maximum and minimum intensity, size and shape.

Among all the statistical approaches one of the most adopted provides the definition of the *co-occurrence matrices* at the gray level: it is based on the estimation of the statistics of the second order of the spatial arrangement of the gray level values.

A co-occurrence matrix [25] is a square matrix in which the elements represent the relative frequency of occurrence of pairs of gray level values of pixels separated by a certain distance in a given direction. Formally, the elements of a co-occurrence matrix can be defined as:

$$C_d(g_1, g_2) = \left| \begin{cases} (a,b) \in N \times N \colon I(a,b) = g_1 \\ I(a+dx, b+dy) = g_2 \\ (a+dx, b+dy) \in N \times N \end{cases} \right| \tag{13}$$

where $I(a,b)$ denotes a square image with a fixed number on gray values, g_1 and g_2 are two gray levels of interest and $|\cdot|$ is the cardinality of a set. Important descriptors are obtained from co-occurrence matrix. One of the most relevant is R_M , often used to measures the smoothness or the homogeneity of an image. It is defined as:

$$R_M = \sum_{g_1} \sum_{g_2} C^2\,(g_1, g_2) \tag{14}$$

where $C(g_1, g_2)$ is a normalized co-occurrence matrix.

As previously stated, in Oriental Medicine different areas of tongue reflect the functioning of internal organs (Fig. 1). Therefore, to intercept possible pathological changes, it is possible to refer to the analysis of different areas of the tongue like Tip, Center, Root, and both Left and Right edge of the tongue. In each area it is then possible to investigate more portions of pixels to obtain a more in-depth sampling. In the hypothesis of reasoning on a single Region of Interest (ROI) for a single inspection area, it is possible to use (14) and (15) to define a set of texture measures.

Starting from co-occurrence matrix it is possible to obtain the spatial gray-tone dependency matrix $G = [p(i,j)]$, where each $[p(i,j)]$ is evaluated considering the mean $p(i,j;d,\theta)$, i.e. the probability of a pixel pair from gray tone i to j with distance d and direction specified by the angle θ [26].

Texture features are relevant in identifying dusty coating of a tongue. Visually, a dusty coating is uniform, smooth on the surface with slow transitions of gray tones. Distinct areas of the tongue may be different in texture, but also in color: a dusty coating often shows white and yellow colors. More formally, given the spatial gray-tone dependency matrix and the quantized gray levels of an image, N_g, texture features are analyzed by evaluating specific parameters such as: the angular second moment, the contrast, the correlation, the variance and the entropy.

Angular Second Moment:

$$ASM = \sum_{i}^{N_g} \sum_{j}^{N_g} \{p(i,j)\}^2 \tag{15}$$

Contrast:

$$C = \begin{cases} \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\} \\ \qquad |i-j| = n \end{cases} \tag{16}$$

Correlation:

$$Corr = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i,j) p(i,j) - \mu_x \cdot \mu_y}{\sigma_x \cdot \sigma_y} \tag{17}$$

Where μ_x μ_y represent the mean of p_x e p_y and σ_x e σ_y represent the standard deviations of p_x e p_y.

Variance:

$$Var = \sum_{i}^{N_g} \sum_{j}^{N_g} (i - u_x)^2 \ p(i,j) \tag{18}$$

Entropy:

$$E = \sum_{i}^{N_g} \sum_{j}^{N_g} p(i,j) \log p(i,j) \tag{19}$$

Therefore, human perception detects the messy coating by observing the properties of fine and invariant surface texture.

Conversely, with Computerized Tongue Diagnosis Systems the dusty coating can be detected by smooth transitions of gray tones. The pixels appear to be highly correlated and characterized by low contrast values in shades of gray.

4.3 Color Features

Human diagnosis can be effectively supported by the automatic analysis of images [28] that is in charge of extracting appropriate features, such as color. The representation of color features can be made in different color spaces [30]. Color space is a method by which color can be displayed. Typically, for automatic image analysis, the most used color spaces include RGB, HSI, CIExy, CIELUV and CIELAB.

Usually, a color is defined by means of three parameters describing the position of the color within the adopted color space. An image is commonly represented with a two-dimensional pixel matrix in which each pixel is composed of three parameters, i.e. red, green and blue (RGB). The RGB color space is often used in computer applications as no transformation for the screen display is needed. Known the parameters of a color in a space, the representation in other color spaces can be obtained through appropriate transformations, see [31] for further details.

As an example, HSI color model uses hue, saturation, and intensity to describe the features inside the images. More specifically, given a generic pixel or a ROI of an image, Hue (H) is the color type of the pixel, Saturation (S) is the degree to which a certain color is mixed into other colors, and Intensity (I) is the brightness of the considered pixel.

The representation in the HSI space can be obtained starting from the RGB one, by applying the following transformations:

$$\mathrm{H} = \arccos \left\{ \frac{[(R-G)+(R-B)]/2}{[(R-G)^2+(R-B)(G-B)]^{1/2}} \right\} \tag{20}$$

$$S = 1 - \frac{3}{(R+G+B)} [min(R,G,B)] \tag{21}$$

$$I = \frac{(R+G+B)}{3} \tag{22}$$

The human eye perceives light through three types of conical cells that intercept the peaks of spectral sensitivity of the wavelengths respectively in the short ("S", 420 nm - 440 nm), medium ("M", 530 nm - 540 nm) and long field ("L", 560 nm - 580 nm). These values represent the so-called tristimulus: this triple can represent the quantity of three primary colors in a three-color and additive color model. The CIE XYZ color space represents the color as visible to the human eye, so CIE XYZ is a device-invariant representation of color. When judging the brightness of different colors, humans tend to perceive light within the green parts of the spectrum as brighter than red or blue light of equal power.

The CIE model exploits this assumption by setting Y as luminance. Z is almost equal to blue, or the response of the cone S, and X is a mix of response curves chosen to be non-negative. The Y setting for luminance has the useful result that for each given Y value, the XZ plane will contain all the possible chromaticity at that luminance.

It is well known that color tongue gives important information about the possible presence of specific pathologies [15, 27]. Various approaches have been proposed in order to exploit color features in tongue medical images. One of the most important approach has been proposed in [33, 34]. The technique defines a kind of color codebook in which the colorimetric features of the images associated with various diseases are mapped. In more details, color features from each pixel are extracted and assigned to 1 of 12 colors symbolizing the tongue color gamut. The tongue color gamut represents all possible visible colors on the tongue surface. Pixels or ROI of the tongue image, will be associated to a vector composed of 12 features, which can be used in the subsequent classification phase. Figure 7, taken from [33], shows within the red boundary, the CIE chromaticity diagram and highlights the black boundary in which lies the 98% of the points.

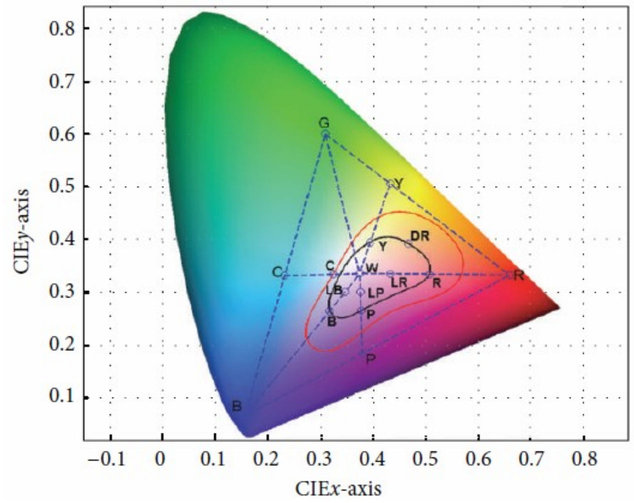


Figure 7: The 100% and 98% tongue color gamut in CIE color space [33]

Starting from these considerations the colors defining the gamut have been identified (Fig.8).

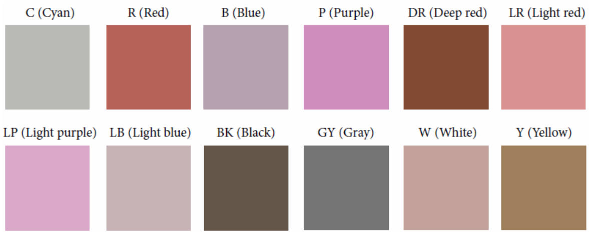


Figure 8: 12 colors representing the tongue color gamut [33]

In addition to the approach that involves the use of color tongue gamut, it's possible to evaluate the mean and standard deviation of a selected ROI. Some recent proposals have highlighted how Multiple Instance Learning approaches are able to obtain interesting classification performances in applications on medical images [35, 37]. In [36] the authors proposed a mixed integer nonlinear formulation solved with MIL approach; the proposed algorithm was applied to a set of color images (Red, Green, Blue, RGB) with the objective of classifying the images containing specific pattern. Multi-instance learning (MIL) is a recent machine learning paradigm that is proving suitable for analyzing medical images and videos. MIL algorithms detect relevant patterns in images or videos based only on the labels of classes globally assigned to images or videos. Therefore, supervision is based on global labels, and the training phase of MIL algorithms does not require tedious manual segmentations. Proposals based on MIL approaches are attracting increasing interest from the Medical Image and Video Analysis (MIVA) community. The need for tools allowing to construct predictive models capturing disease progression is a priority [44]. For the same reason, great attention is devoted to effective solutions ensuring accuracy in the identification of groups of similar genes or patients [45] and capturing valuable insights from medical sources [46].

5. Conclusion and Future Work

In this paper, we focused on the features most commonly used to perform automatic analysis of tongue images. This work has been motivated by the observation that in many research works on image processing in medical field, features are often undefined and a clear formulation of used objective methods is missing. Our aim was to give a contribution to tongue image analysis and classification by an in depth analysis of relevant features.

Using appropriate features allows, through approach like Artificial Neural Networks (ANN) and Support Vector Machines (SVM), to perform an effective automatic classification between images of healthy tongue and of tongue with pathologies [38]. The use of ANN seems useful to propose more general model to map health data [39]. Therefore, it is quite evident that reasoning about the most relevant and useful features for automatic classification methods is a mandatory step.

In computer vision systems, the evolution of camera technology even on wearable devices opens to the possibilities of creating self-diagnosis systems. Many different sophisticated systems working on imaging analysis have been proposed in the recent literature in different areas, such as health care [40, 41]. As for future work, the aim is to further investigate relevant features useful for automatic classification methods in specific domain, and take advantage of machine learning technique for an effective classification. More specifically, the authors plan to perform a detailed comparison of the approaches in the literature and provide insights on the best features predictors.

ACKNOWLEDGMENTS

A special thanks to Dr. Fulvio Limido, professor of homeopathy at the SBM - Italian Biotherapeutic Medical Society, for the support in understanding the medical topics treated.

REFERENCES

[1] Y. Q. Liu, Y. X. Wang, N. N. Shi, X. J. Han, and A. P. Lu, 2016. Current situation of International Organization for Standardization/Technical Committee 249 international standards of traditional Chinese medicine. Chinese Journal of Integrative Medicine, vol. 2016, pp. 1–5.

[2] T. Kawanabe, N. D. Kamarudin, C. Y. Ooi et al., 2016. Quantification of tongue color using machine learning in Kampo medicine. European Journal of Integrative Medicine, vol. 8, no. 6, pp. 932–941.

[3] X. Wang, B. Zhang, Z. Yang, H. Wang, and D. Zhang, 2013. Statistical analysis of tongue images for feature extraction and diagnostics. IEEE Transactions on Image Processing, vol. 22, no. 12, pp. 5336–5347.

[4] X. Wang and D. Zhang, 2010. An optimized tongue image color correction scheme. IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 6, pp. 1355–1364.

[5] B. Pang, D. Zhang, N. Li, and K. Wang, 2004. Computerized tongue diagnosis based on Bayesian networks. IEEE Transactions on Biomedical Engineering, vol. 51, no. 10, pp. 1803–1810.

[6] B. Zhang, X. Wang, J. You, and D. Zhang, 2013. Tongue color analysis for medical application. Evidence-Based Complementary and Alternative Medicine, vol. 2013, Article ID 264742, p. 11.

[7] T. T. Deng, 1995. Basic theory of traditional Chinese medicine. In Diagnostics of Chinese Medicine, Chih-Yin Publishing, Taipei, Taiwan, pp. 5–11.

[8] C. H. Li and P. C. Yuen, 2002. Tongue image matching using color content. Pattern Recognition, vol. 35, no. 2, pp. 407–419.

[9] B. Pang, D. Zhang, and K. Wang, 2005. Tongue image analysis for appendicitis diagnosis. Information Sciences, vol. 175, no. 3, pp. 160–176.

[10] Y.G. Wang, J. Yang, Y. Zhou, and Y.Z. Wang, 2007. Region partition and feature matching based color recognition of tongue image. Pattern Recognition Letters, vol. 28, no. 1, pp. 11–19.

[11] C.C. Chiu, 2000. A novel approach based on computerized image analysis for traditional Chinese medical diagnosis of the tongue. Computer Methods and Programs in Biomedicine, vol. 61, no. 2, pp. 77–89.

[12] Z. Guo, 2008. Tongue image matching using color and texture. In Proceedings of the International Conference on Medical Biometrics (ICMB '08), pp. 273–281.

[13] C. Liu, Z. Han, and T.Xie, 2015. Hyperspectral high-dynamic-range endoscopic mucosal imaging. Chin. Opt. Lett. 13, 071701.

[14] C.C. Chiu, 1996. Development of a computerized tongue diagnosis system", Biomedical Engineering—Applications, Basis and Communications, vol. 8, no. 4, pp. 342–350.

[15] B. Pang, D. Zhang, and K. Wang, 2005. The bi-elliptical deformable contour and its application to automated tongue segmentation in Chinese medicine. IEEE Transactions on Medical Imaging, vol. 24, no. 8, pp. 946–956.

[16] N. M. Li, 1994. The Contemporary Investigations of Computerized Tongue Diagnosis. The Handbook of Chinese Tongue Diagnosis, Shed-Yuan Publishing, Beijing, China.

[17] N. Li, D. Zhang, and K. Wang, 2006. Tongue Diagnostics. Shed-Yuan Publishing, Beijing, China.

[18] E. K. Pae and A. A. Lowe, 1999. Tongue shape in obstructive sleep apnea patients. Angle Orthodontist, vol. 69, no. 2, pp. 147–150.

[19] B. Huang, J. Wu, D. Zhang, and N. Li, 2010. Tongue shape classification by geometric features. Information Sciences, vol. 180, no. 2, pp. 312–324.

[20] R. O. Duda, P. E. Hart, and D. G. Stork, 2000. Pattern Classification. Wiley-Interscience, New York, NY, USA, 2nd edition, ISBN: 978-0-471-05669-0.

[21] G. Anitha, M. Ismail and U. Ahmed, 2017. Analysis of Diabetes Mellitus and NPDR through Characterization of Changes in Tongue Geometry Features using NN Classifier. International Journal of Pure and Applied Mathematics, volume 117 No. 15 2017, pp. 1015-1019 ISSN: 1311-8080 (printed version)/1314-3395 (on-line version).

[22] Bob Zhang and Han Zhang, 2015. Significant Geometry Features in Tongue Image Analysis. Evidence-Based Complementary and Alternative Medicine, vol. 2015, Article ID 897580, 8 pages. https://doi.org/10.1155/2015/897580.

[23] Bo Pang, D. Zhang, N. Li and Kuanquan Wang, 2004. Computerized tongue diagnosis based on Bayesian networks. In IEEE Transactions on Biomedical Engineering, vol. 51, no. 10, pp. 1803-1810. doi: 10.1109/TBME.2004.831534

[24] H. Z. Zhang, K. Q. Wang, D. Zhang, B. Pang and B. Huang, 2005. Computer Aided Tongue Diagnosis System. *IEEE Engineering in Medicine and Biology 27th Annual Conference*, Shanghai, pp. 6754-6757. doi: 10.1109/IEMBS.2005.1616055.

[25] R. M. Haralick, K. Shanmugan, and I. Dinstein, 1973. Textural features for image classification. In IEEE Trans. Syst., Man, Cybern., vol. SMC-3, pp. 610–621.

[26] G. Maciocia, 1995. Tongue Diagnosis in Chinese Medicine. Seattle, WA: Eastland.

[27] B. Li, Q. Huang, Y. Lu, S. Chen, R. Liang, and Z. Wang, 2007. A method of classifying tongue colors for traditional Chinese medicine diagnosis based on the CIELAB color space. In Proceedings of the International Conference on Medical Biometrics, pp. 153–159.

[28] C. C. Chiu, 1996. Development of a computerized tongue diagnosis system. Biomedical Engineering, vol. 8, no. 4, pp. 342–350.

[29] B. Kirschbaum, 2000. Atlas of Chinese Tongue Diagnosis, Eastland Press, Seattle, Wash, USA.

[30] I. Pita, 1993. Digital Image Processing Algorithms. Englewood Cliffs, NJ: Prentice-Hall, pp. 23–40.

[31] R. Kanawong, T. Obafemi-Ajayi, T. Ma, D. Xu, S. Li, and Y. Duan, 2012. Automated Tongue Feature Extraction for ZHENG Classification in Traditional Chinese Medicine. Evidence-Based Complementary and Alternative Medicine, vol. 2012, Article ID 912852, 14 pages. https://doi.org/10.1155/2012/912852.

[32] D. Zhang, 2000. Automated Biometrics: Technologies and Systems. Kluwer Academic Publisher, Boston, Mass, USA.

[33] Bob Zhang, Xingzheng Wang, Jane You, and David Zhang, 2013. Tongue Color Analysis for Medical Application. Evidence-Based Complementary and Alternative Medicine, vol. 2013, Article ID 264742, 11 pages. https://doi.org/10.1155/2013/264742.

[34] X. Wang and D. Zhang, 2011. Statistical tongue color distribution and its application. In Proceedings of the International Conference on Computer and Computational Intelligence.

[35] A. Astorino, A. Fuduli, M. Gaudioso and E. Vocaturo, 2018. A Multiple Instance Learning Algorithm for Color Images Classification. In Proceedings of IDEAS 2018, Villa San Giovanni, Italy, pp. 262-266, https://doi.org/10.1145/3216122.3216144.

[36] G. Quellec, G. Cazuguel, B. Cochener, M. Lamard, 2017. Multiple-instance learning for medical image and video analysis. IEEE Reviews in Biomedical Engineering 10, pp. 213-234.

[37] A. Astorino, A. Fuduli, P. Veltri and E. Vocaturo, 2017. On a recent algorithm for multiple instance learning. Preliminary applications in image classification. In: 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1615-1619.

[38] Y. Zhang, R. Liang, Z.Wang, Y. Fan, and F. Li, 2005. Analysis of the color characteristics of tongue digital images from 884 physical examination cases. Journal of Beijing University of Traditional Chinese Medicine, vol. 28, pp. 73–75.

[39] E. Vocaturo, P. Veltri, 2017. On the use of networks in biomedicine. In: 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017), July 24-26, 2017, Leuven, Belgium, pp. 498- 503, ISSN 1877-0509, https://doi.org/10.1016/j.procs. 2017.06.132.

[40] E. Zumpano et al., 2018. SIMPATICO 3D: A Medical Information System for Diagnostic Procedures. IEEE International Conference on Bioinformatics and Biomedicine, BIBM Madrid 2018, pp. 2125-28, http://doi.ieeecomputersociety.org/10.1109/BIBM.2018.8621090.

[41] H. Adams, J. Shinn, W. G. Morrel, J. Noble, B. Bodenheimer, 2019. Development and evaluation of an immersive virtual reality system for medical imaging of the ear. Vol. 10951.

[42] E. Vocaturo, E. Zumpano, P. Veltri, 2018. Image pre-processing in computer vision systems for melanoma detection. In: IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018, Madrid, Spain, pp. 2117-2124.

[43] E. Vocaturo, E. Zumpano, P. Veltri, 2018. Features for melanoma lesions characterization in computer vision systems. In: 9th International Conference on Information, Intelligence, Systems and Applications, IISA 2018, Zakynthos, Greece, pp. 1-8.

[44] L. Caroprese, P. Veltri, E. Vocaturo, E. Zumpano, 2018. Deep Learning Techniques for Electronic Health Record Analysis. IISA 2018 pp. 1-4.

[45] E. Masciari, G. M. Mazzeo, C. Zaniolo, 2014. Analysing microarray expression data through effective clustering. Inf. Sci. 262: 32-45.

[46] S. Greco, C. Molinaro, I. Trubitsyna, 2018.Computing Approximate Query Answers over Inconsistent Knowledge Bases. IJCAI 2018, pp.1838-1846.

Differentially Private Sequential Pattern Mining considering Time Interval for Electronic Medical Record Systems

Hieu Hanh Le
Tokyo Institute of Technology
Tokyo, Japan
hanhlh@de.cs.titech.ac.jp

Muneo Kushima
University of Miyazaki Hospital
Miyazaki, Japan
muneo_kushima@med.miyazaki-u.ac.jp

Kenji Araki
University of Miyazaki Hospital
Miyazaki, Japan
taichan@med.miyazaki-u.ac.jp

Haruo Yokota
Tokyo Institute of Technology
Tokyo, Japan
yokota@cs.titech.ac.jp

ABSTRACT

Electronic medical record (EMR) systems have now been widely adopted to support medical workers. There also has been much interest in the machine-based generation of clinical pathways that can utilize sequential pattern mining (SPM) to extract them from historical EMR systems. However, the existing methods do not protect individual privacy, even though they involve sensitive medical data. To ensure the privacy of individual data, this paper describes two algorithms that deploy differential privacy by adding noise during calculations in the SPM considering time interval for guaranteeing privacy. The proposals can limit the amount of added noise by adding noise to the frequency calculations of only a part of candidate closed sequences. Experiments on real medical datasets show that our proposal can ensure the robust and high utility of mining process even with minimum privacy budget and amount of added noise.

CCS CONCEPTS

• **Information systems → Data mining**; • **Security and privacy → Privacy protections**.

KEYWORDS

Electronic medical record, typical pathway, differential privacy, sequential pattern mining

ACM Reference Format:
Hieu Hanh Le, Muneo Kushima, Kenji Araki, and Haruo Yokota. 2019. Differentially Private Sequential Pattern Mining considering Time Interval for Electronic Medical Record Systems. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3331076.3331098

1 INTRODUCTION

Electronic medical record (EMR) systems have now been adopted by most large-scale hospitals to support medical workers via simple record-keeping procedures. In addition, the secondary uses of EMR data have attracted attention with respect to the standardization of medical treatments. Medical workers tend to use clinical pathways as guidelines for their medical actions. A clinical pathway defines the typical flow of medical treatment for each disease and is generated conventionally by medical workers drawing on their experience.

There have been several methods for the machine-based generation of clinical pathways that can utilize sequential pattern mining (SPM) to extract the pathways from historical EMR systems [9, 10, 14]. In our previous work, we proposed a method called T-CSpan that extracts frequent sequential patterns from EMR systems to generate clinical pathways automatically with handling time intervals and the efficacy of medicines [10].

However, releasing clinical pathways risks compromising individual patient privacy. Due to the changes in new discovered medicines or medical equipment, the typical clinical pathways may change as well. Therefore, malicious users, who can get the different outputs of the algorithm via changing the search condition, may identify individual's records. Some approaches to this issue have considered anonymizing the data before the mining operations [11, 15]. However, it has been shown that even anonymized data can become reidentifiable via public data or other data mining algorithms [7, 13].

An alternative approach proposes differential privacy as a way to address such issues efficiently [5]. It offers strong theoretical guarantees on the privacy of released data by adding a carefully chosen amount of noise to the analyzed results. Differential privacy ensures that the output of computation is insensitive to a change in any individual's record, thereby restricting privacy leaks from the results. Although there have been a number of existing works on differential privacy SPM [2, 17], however they only consider dataset of numerical data which is much simpler than EMR data that consists of complex medical-orders such as prescription, injection, surgery, etc.

In this paper, we focus on the design of a differentially private frequent-sequence mining algorithm considering time interval for

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331098

EMR data. Given a collection of input medical-order sequences, the algorithm aims to find those sequences that occur in the collection more frequently than a given threshold (minimum support or *MinSup*). During the calculations that generate the frequent sequences, noise elements are carefully added with the aim of avoiding significant degradation of the algorithm's utility. Taking these advantages into consideration, we decide to deploy differential privacy to T-CSpan, realizing that there are significantly fewer calculations in T-CSpan than in other relevant algorithms as T-CSpan calculates only the frequencies of closed sequences [10]. Because a closed sequence has no super-sequences with the same frequencies, all sequences of it can be ignored during calculation. Moreover, only closed sequences that have true frequency closed to the *MinSup* are considered as candidate sequences to which noise is added. Thus, the required amount of added noise can be remarkably restrained.

The contribution of this paper are as follows.

- We propose two algorithms called T-CSpan-DP and an enhanced version T-CSpan-DPe to achieve the goal of providing a SPM algorithm considering time interval with satisfying differential privacy for EMR systems.
- The proposed algorithms are evaluated using real medical data. The experimental results identify the better algorithm T-CSpan-DPe with respect to high utility and small amount of added noise.
- It is demonstrated that T-CSpan-DPe's performance is robust as it is not affected by the parameters controlling privacy budget and noise used in the algorithms. As a result, the proposed algorithm has a high potential to be applied to private typical-pathway-generation from EMR system because it can maintain high utility with low privacy budget and a small amount of noise.

The remainder of this paper is organized as follows. Related work is reviewed in Section 2. The key definitions and theorems of differential privacy are introduced in Section 3. The proposed method is described in Section 4. Experimental evaluation of the proposed method is discussed in Section 5. Conclusions and ideas for future work are summarized in Section 6.

2 RELATED WORK

This section gives a brief review of sequential pattern mining (SPM), and SPM with differential privacy.

2.1 Sequential Pattern Mining

A well-known SPM algorithm is a Priori-based frequent pattern-mining algorithm [1]. However, it is very time-consuming with large data sets and generates many irrelevant patterns among its results. To exclude irrelevant patterns, PrefixSpan [8] was proposed to mine the complete set of patterns while reducing the effort of candidate pattern generation by exploring prefix projection. To improve efficiency further, CSpan [12] was proposed for mining closed sequential patterns. This algorithm uses a pruning method called occurrence checking that allows the early detection of closed sequential patterns during the mining.

Initially, the proposed method of Agrawal et al. [1] did not consider the time interval between items. For example, the injection was performed on January 1, 2019, the sequence for performing surgery the next day, and the sequence for performing surgery three days after the injection was regarded as the same sequence. Chen et al. proposed a mining method called TI-SPM for sequences where the time interval is important, such as medical instructions, which should treat the above two sequences as different things [4].

T-PrefixSpan [14] is a method to extract frequent sequential patterns from EMR logs that considers time intervals and the efficacy of medicines. Moreover, because time intervals were incorporated, T-PrefixSpan is functionally rich because it can offer medical workers more valuable information about the minimum, maximum, average, median, and most frequent value of time intervals between successive medical treatments. T-CSpan [10] further improves the speed performance by applying the idea of mining only closed sequences.

2.2 Differentially Private Sequential Pattern Mining

Differential privacy is now considered the standard approach to private data analysis [5, 6]. It has been shown to be resistant to composition attacks, in which an adversary uses independent anonymized data to breach privacy [7]. Detailed statistical aspects of differential privacy have also been studied [16].

There are several works on differentially private frequent-sequence mining. Bonomi et al. [2] propose a two-phase differentially private algorithm for mining frequent consecutive-item sequences. The first phase utilizes a prefix tree to find candidate sequences, then leverages a database-transformation technique to refine the support of candidate sequences. Xu et al.[17] utilize sample databases to estimate the sequences that are potentially frequent, then reduce the number of candidate sequences. However, all existing works consider only simple datasets of numerical data. In our work, the dataset contains longer sequences of more complex items, such as prescriptions, and surgical procedures. Moreover, existing works have focused on Top-k mining in which the number of output results is fixed, hence the amount of added noise can be easily controlled. However, in our work, the number of output results are varied and depends on a predefined threshold *MinSup*. The extracted sequences have the frequencies greater than *MinSup*. Therefore, increasing *MinSup* will decrease the number of extracted sequences and vice versus.

Our work is based on T-CSpan [10], which is the fastest algorithm for generating clinical pathways that considers the time interval between successive items in EMRs. T-CSpan mines closed sequential patterns using an occurrence-checking method that excludes duplicated sequences in the mining process.

3 PRELIMINARIES

Differential privacy has become a de facto standard for privacy considerations in private data analysis [5, 6]. Formally, differential privacy is defined as follows.

Definition 1. ***ϵ-differential privacy***
A private algorithm M satisfies ϵ-differential privacy if and only if, for any databases D_1 and D_2 that differ by at most one record and for any subset of output S,
$Pr\left[M(D_1) \in S\right] \leq exp(\epsilon) \times Pr\left[M(D_2) \in S\right]$,
where the probability is taken over the randomness of M.

An essential concept for guaranteeing differential privacy is the sensitivity. It is used to measure the maximum change in the outputs of a function when any individual's record in the database is changed.

Definition 2. ***sensitivity*** Δf
Given any function $f : D \to R^n$ for any databases D_1 and D_2 that differ by at most one record, the sensitivity of function f is $\Delta f = max_{D_1,D_2} \|f(D_1) - f(D_2)\|$.

For example, for the function of counting the male students in a class, the sensitivity becomes 1.

The Laplace mechanism has been proposed [5]. And it is proven that differential privacy can be achieved by adding noise drawn randomly from Laplace distribution.

Theorem 1. ***Laplace mechanism***
For any function $f : D \to R^n$ with sensitivity Δf, the algorithm
$M(D) = f(D) + Lap(\lambda)$
satisfies ϵ-differential privacy, where $Lap(\lambda)$ follows the probability density function $Pr[x|\lambda] = \frac{1}{2\lambda} exp(-\frac{|x|}{\lambda})$, where $\lambda = \frac{\Delta f}{\epsilon}$.

Sequential composition and parallel composition are used for supporting multiple differentially private calculations[6].

Theorem 2. ***Sequential composition***
Let M_1, M_2, ..., M_k be k algorithms, each provides ϵ_i-differential privacy. A sequence of algorithms $M_i(D)$ over database D provides $(\sum_i \epsilon_i)$-differential privacy ($i = 1, ..., k$).

Theorem 3. ***Parallel composition***
Let M_1, M_2, ..., M_k be k algorithms, each provides ϵ_i-differential privacy. A sequence of algorithms $M_i(D_i)$ over disjoint databases D_i, ..., D_k provides $max(\epsilon_i)$-differential privacy ($i = 1, ..., k$).

4 PROPOSED METHODS

First, we define the concepts necessary for the introduction of the proposed algorithm. We then explain the algorithms that extract typical sequences while achieving differential privacy. Dealing with medicines and their efficacy and other detailed information are discussed in [10].

4.1 Handling Time Intervals between Items

Definition 3. ***T-item*** (i, t)
*Let I be a set of items and t be the time that item i occurs. The **T-item** (i, t) is defined as the pair comprising i and t.*

Definition 4. ***T-sequence*** *s and* ***O-sequence*** O_s
*Let **T-sequence** s be a sequence of T-items such that*

$$s =< (i_1, t_1), (i_2, t_2), ..., (i_n, t_n) > .$$

*T-items that occur at the same time shall be arranged in dictionary order. Furthermore, if n is the length of T-sequence s, the **O-sequence** of s is defined as the sequence $O_s =< i_1, i_2, ..., i_n >$.*

Definition 5. ***time-interval*** TI_k
*Given a T-sequence $s =< (i_1, t_1), (i_2, t_2), ..., (i_n, t_n) >$, let **time-interval** TI_k be defined as*

$$TI_k \equiv t_{k+1} - t_k \quad (k = 1, 2, ..., n - 2, n - 1).$$

Definition 6. ***T-sequential database*** *D and* ***O-sequential database*** O_D
*Given a set of T-sequences, a **T-sequential database** D is defined as*

$$D \equiv \{(s_{id}, s) \mid s_{id}, s \in S\},$$

*where the identifier s_{id} for the elements of D is unique to each sequence. Furthermore, let an **O-sequential database** O_D be a sequential database comprising the O-sequences configured from all T-sequences in D. Let $Size(D)$ be $Size(O_D)$, i.e., the number of sequences in O_D.*

Definition 7. ***T-frequent sequential pattern*** *P*
Let MinSup ($0 \leq MinSup \leq 1$) be a minimum support and D be a T-sequential database. Given $P =< i_1, X_1, i_2, X_2, ..., i_{n-1}, X_{n-1}, i_n >$ (where $\forall j$ i_j is an item and $\forall k$ X_k is the set of five values: min_k, mod_k, ave_k, med_k and max_k), a sequence $O_P =< i_1, i_2, ..., i_{n-1}, i_n >$ can be configured.
*P is defined as a **T-frequent sequential pattern** if O_p is a frequent sequential pattern in an O-sequential database configured from D (i.e., $Sup(P) =| \{Seq|O_p \subseteq Seq, (s_{id}, Seq) \in O_D$, where s_{id} is an identifier of $Seq\} |\geq Size(O_D) \times MinSup$).*
Let O_P be the O-Pattern of P. The set of five values are defined as a result of the following considerations.
Given all T-sequences for which the O-sequences contain O_P in D, let S be one such T-sequence, with $S =< i'_1, t_1, i'_2, t_2, .., i'_{m-1}, t_{m-1}, i'_m >$. By using $j_1, j_2, ..., j_{n-1}, j_n$, which fulfil both $1 \leq j_1 < j_2 < ... < j_{n-1} < j_n \leq m$ and $i_k = i'_{j_k}, i_{k+1} = i'_{j_{k+1}}$, sets of time-intervals can be configured: $Set_{TI_1}, Set_{TI_2}, ..., Set_{TI_{n-1}}$, where $TI_k = t'_{j_{k+1}} - t'_{j_k}$. In $X_k = (min_k, mod_k, ave_k, med_k, max_k)$, the five values can be defined as

1. $min_k = \min Set_{TI_k}$
2. mod_k = *the most frequent value in* Set_{TI_k}
3. ave_k = *the average of values in* Set_{TI_k}
4. med_k = *the intermediate value of values in* Set_{TI_k}
5. $max_k = \max Set_{TI_k}$

Definition 8. ***T-closed frequent sequential pattern*** *A*
*Given a T-sequential database D, let $\sum$ be the set of T-frequent sequential patterns extracted from D and let A be a T-frequent sequential pattern in $\sum$. A is a **T-closed frequent sequential pattern** if there is no B in $\sum \backslash A$ that*

1. *If A' and B' are the O-Patterns of A and B, respectively, then $A' \subseteq B'$.*
2. *$Sup(A) \leq Sup(B)$, where we define the support for a T-frequent sequential pattern A as $Sup(A) \equiv| \{s|s \subseteq S, (s_{id}, S) \in D$ and s_{id} is the identifier of S in $D\} |$.*
3. *If A and B are $< a_1, T_1, a_2, T_2, ..., a_{n-1}, T_{n-1}, a_n >$ and $< b_1, T'_1, b_2, T'_2, ..., b_{m-1}, T'_{m-1}, b_m >$, respectively, then $< j_1, j_2, ..., j_n >$ that meets $1 \leq j_1 < j_2 < ... < j_n \leq m$ and $a_k = b_{j_k}, a_{k+1} = b_{j_{k+1}}$ exists.*

For example, consider extracting T-frequent sequential patterns from a T-sequential database such as the one described in Table 1 under the minimum support $MinSup = 0.4$. The O-sequential database of D is shown in Table 2. The frequent sequential patterns on the minimum support $MinSup = 0.4$ are $< a >, < b >, < d >, < a, b >, < b, d >$ and $< a, b, d >$. Because the frequent sequential patterns that have one item in O_D are T-frequent sequential patterns

Table 1: T-sequential database D

Sequence identifier	T-sequence
s_1	$< (a,1),(b,3),(c,7),(d,10) >$
s_2	$< (a,1),(b,4),(d,7) >$
s_3	$< (a,2),(b,6),(b,9) >$
s_4	$< (a,2),(b,5) >$
s_5	$< (a,2),(b,7) >$

Table 2: O_D (O-sequential database of D)

Sequence identifier	T-sequence
s_1	$< a,b,c,d >$
s_2	$< a,b,d >$
s_3	$< a,b,b >$
s_4	$< a,b >$
s_5	$< a,b >$

in D; $< a >$, $< b >$, and $< d >$ are T-frequent sequential patterns in D. Considering the time between item a and item b in the sequence $< a, b >$, the set of time intervals calculated from D is {2, 3, 3, 4, 5}. Considering the minimum, the most frequent value, the average, the median, and the maximum, $< a$, (2, 3, 3, 3, 5), $b >$ is a T-frequent sequential pattern in D ($2+3+3+4+5 = 17$, $[17/5] = 3$). Similarly, if we calculate T-frequent sequential patterns from $< b, d >$ and $< a, b, d >$, these are $< b$, (3, 5, 5, 5, 7), $d >$ and $< a$, (2, 2, 2, 2, 3), b, (3, 5, 5, 5, 7), $d >$. If more than two different values are the most frequent values, their average is the most frequent value. Therefore, T-frequent sequential patterns in D under the minimum support *MinSup* = 0.4 are $< a >$, $< b >$, $< d >$, $< a$, (2, 3, 3, 3, 5), $b >$, $< b$, (3, 5, 5, 5, 7), $d >$, and $< a$, (2, 2, 2, 2, 3), b, (3, 5, 5, 5, 7), $d >$, and T-closed frequent sequential patterns are $< a >$, $< b >$, $< a$, (2, 3, 3, 3, 5), $b >$ and $< a$, (2, 2, 2, 2, 3), b, (3, 5, 5, 5, 7), $d >$.

4.2 T-CSpan-DP Algorithm

T-CSpan-DP is the following **Algorithm 1**, for which the O-sequential database of a T-sequential database D is O_D, the O-sequence of a T-sequence S is O_S, and the connection of a sequence A to a sequence B is AB. Denote the N-th element of a set X as X_N, the N-th item of a sequence S as S_N and the time of occurrence for the N-th T-item of T-sequence A as T_{A_N}.

In this paper, we represent an item in the form of a set of four segments of text (*Class*; *Description*; *Code*; *Name*). *Class* is the type of medical treatment, *Description* is the detailed record of the treatment, *Code* is a medicinal code representing the unique efficacy of the medicine used, and *Name* is the name of the medicine. An example of such an item is (*prescription*; *internal medicine*; *613*; *Fine Cefzon 10%*). We then configure a T-sequence that comprises the T-items undergone by one patient until discharge from hospital and construct a T-sequential database from these T-sequences.

The process in T-CSpan-DP is efficient because it utilizes an occurrence check only to add the closed T-sequences to the result. A closed T-frequent sequential pattern is a frequent sequence that has no super-sequence with the same support. Therefore, only the longest frequent T-sequences need to be added, with any sequences of it being ignored unless they occur more frequently than the super-sequence.

To meet the differential privacy requirement, noise is added to the true frequency of the generated closed T-sequence, as specified in line 5 of **Algorithm 2**. The noise is generated according to a Laplace distribution. Furthermore, to minimize the added noise, we only add noise to closed sequences that have a frequency close to the *MinSup*. The implicit idea is that as we added noise to the true frequency of the sequences, some of the true frequent sequences may be excluded from the outputs. Hence, we include the sequences that have the true frequencies closed to the *MinSup* into consideration. It is achieved by introducing the β parameter. The number of such sequences is controlled via the β parameter at line 4 of **Algorithm 2**.

THEOREM 4. ***Differential privacy*** *T-CSpan-DP satisfies $l \times \epsilon$-differential privacy, where l is the number of times* ***Algorithm 2*** *is invoked.*

Proof: In **Algorithm 2**, for the computation of the frequency of sequences, the sensitivity is 1. So, adding noise $Lap(\frac{1}{\epsilon})$ in computing the sequence's frequency in **Algorithm 2** satisfies ϵ-differential privacy. As **Algorithm 2** is invoked by l times from **Algorithm 1** (line 7 to 9), from Sequential composition theorem (Theorem 2), T-CSpan-DP satisfies $l \times \epsilon$-differential privacy. l is proportional with the number of single frequent items and depends on the predefined *MinSup* as the decreasing *MinSup* will increase l as more items can be considered as frequent ones.

4.3 T-CSpan-DPe Algorithm

As T-CSpan-DP algorithm satisfies $l \times \epsilon$-differential privacy, it is still inefficient as the added noise may be large. We further propose an enhance algorithm called T-CSpan-DPe. Instead of adding noise during generating closed sequences as in **Algorithm 4**, we only add noise to closed sequences that were generated after all (**Algorithm 3**, line 10 to 13). We also adjust the number of candidate sequences by using parameter β as in T-CSpan-DP.

THEOREM 5. ***Differential privacy*** *T-CSpan-DPe satisfies ϵ-differential privacy.*

Proof: For the computation of the frequency of sequences, the sensitivity is 1. So, adding noise $Lap(\frac{1}{\epsilon})$ in the closed sequence's frequency in **Algorithm 3** make T-CSpan-DPe satisfies ϵ-differential privacy.

4.4 Proposed Methods at other Systems

Although our proposed algorithms have focused on EMR data, they also can be extended to other types of systems containing private data, where time interval is important. For example, our proposed methods can be applied to the medical insurance claim data which comprise the specifications of medical fees charged to health insurers. So, the trend of prescription while guaranteeing individual privacy can be analyzed. Moreover, the proposed methods also can be applied to retailing data where the habits, interests and the timing of shopping of clients can be studied; or to traveler records at travel agencies where both the places and the time a traveler spent at that places can be derived.

Algorithm 1: T-CSpan-DP

```
Input  : $D$: a T-sequential database, $seq$: a sequence,
         $MinSup$: a minimum support, $\epsilon$: privacy budget, $\beta$:
         range parameter
Output : $P$: the set of T-frequent sequential patterns with
         noise
begin
    $D' |_{seq} = O_{D|_{seq}}$;
    if $seq \, ! = null$ then
        $P \leftarrow$ GetProperTime($seq, D |_{seq}, D' |_{seq}$);
    $SingleFreqItems \leftarrow \{sfi \mid (s \subseteq D' |_{seq}, sfi \in s) \wedge (Sup(sfi) \geq Size(\mathrm{D}) \times MinSup)\}$ ;
    $CSP_{seq} \leftarrow \emptyset$;
    for $sfi \in SingleFreqItems$ do
        $CS_{seq} \leftarrow GenClosedSeqs - DP(D |_{sfi}, sfi, MinSup, \epsilon, \beta)$;
        $CSP_{seq} \leftarrow CSP_{seq} \cup CS_{seq}$;
    for $csp \in CSP_{seq}$ do
        T-CSpan-DP($csp$, $D |_{CSP_{seq}}$, $MinSup$, $\epsilon$, $\beta$);
    Function GetProperTime ($seq$,$D |_{seq}$,$D' |_{\alpha}$)
        if $length(seq) == 1$ then
            return $seq$;
        $K \leftarrow \{k \mid < s_{id}, s > \in D |_{seq}, O_s \in D' |_{seq}, k \subseteq s, O_k == seq\}$ ;
        $T = \{\{\}, \{\}, ..., \{\}\}(\mid T \mid = length(seq) - 1)$ ;
        for $k \in K$ do
            for $i = 0, ..., length(k - 1)$ do
                $T_i \leftarrow T(k_{i+1}) - T(k_i)$;
        $W =< seq_0, seq_1, ..., seq_{length(seq)-1} >$;
        for $i = 0, ..., length(seq) - 2$ do
            $T_i$ = an arbitrary function to exclude outliers from $T_i$;
            $min_i = \min T_i$;
            $mod_i$ =the most frequent value of $T_i$;
            $ave_i$ =the average of values of $T_i$;
            $med_i$ =the intermediate value of $T_i$;
            $max_i = \max T_i$;
            $X_i = (min_i, mod_i, ave_i, med_i, max_i)$;
            $W =< seq_0, ..., seq_i, X_i, seq_{i+1} ..., seq_{length(seq)-1} >$;
        return $W$;
```

Algorithm 2: GenClosedSeqs-DP

```
Input  : $PD$: a projected T-sequential database, $seq$: a
         T-sequence, $MinSup$: a minimum support, $\epsilon$:
         privacy budget, $\beta$: range parameter
Output : $CS$: noisy-added closed T-sequences with prefix $seq$
begin
    $CS_{seq} \leftarrow \emptyset$;
    $F \leftarrow \{f : (s \subseteq PD, f \in s) \wedge (Sup(f) \geq Size(\mathrm{PD}) \times MinSup)\}$;
    if $(Sup(seq) \geq Size(PD) \times MinSup \times (1 - \beta)) \wedge (F \neq \emptyset)$ then
        $noise\_Sup(Seq) = Sup(seq) + Lap(\frac{1}{\epsilon})$;
        if $noise\_Sup(Seq) \geq Size(PD) \times MinSup$ then
            if $seq.closed() == true$ then
                $CS_{seq} \leftarrow seq$;
        for $f \in F$ do
            $CS_{seq} \leftarrow GenClosedSeqs - DP(PD, seq + f, MinSup)$
            $//seq + f$: append $f$ to $seq$;
```

Algorithm 3: T-CSpan-DPe

```
Input  : $D$: a T-sequential database, $seq$: a sequence,
         $MinSup$: a minimum support, $\epsilon$: privacy budget, $\beta$:
         range parameter
Output : $P$: the set of T-frequent sequential patterns with
         noise
begin
    $D' |_{seq} = O_{D|_{seq}}$;
    if $seq \, ! = null$ then
        $P \leftarrow$ GetProperTime($seq, D |_{seq}, D' |_{seq}$);
    $SingleFreqItems \leftarrow \{sfi \mid (s \subseteq D' |_{seq}, sfi \in s) \wedge (Sup(sfi) \geq Size(D) \times MinSup \times (1 - \beta))\}$ ;
    $CSP_{seq} \leftarrow \emptyset$;
    for $sfi \in SingleFreqItems$ do
        $CS_{seq} \leftarrow GenClosedSeqs(D |_{sfi}, sfi, MinSup, \beta)$;
        $CSP_{seq} \leftarrow CSP_{seq} \cup CS_{seq}$;
    for $csp \in CSP_{seq}$ do
        $noise\_Sub(csp) = Sub(csp) + Lap(\frac{1}{\epsilon})$;
        if $noise\_Sub(csp) \geq Size(D)\ timesMinSup$ then
            T-CSpan-DPe($csp$, $D |_{CSP_{seq}}$, $MinSup$, $\epsilon$, $\beta$);
```

5 EXPERIMENTAL EVALUATION

To the best of our knowledge, T-CSpan-DP and T-CSpan-DPe are the first algorithms to support the mining of medical order sequences to generate typical pathways with time interval that guarantee differential privacy. We performed experiments to evaluate the two algorithms with respecting to the amount of added noise and utility using real datasets with a variety of sizes and data distributions. Next, we investigate the effects of ϵ and β on the algorithm's utility.

5.1 Experimental Environment and Method

The algorithms were implemented in Java and evaluated using various minimum support (*MinSup*) values, ϵ and β. Experiments were conducted on a Linux Server with an Intel Xeon CPU E5-4650 at 2.70 GHz and 64 GB RAM.

The target clinical pathway data were medical treatment data recorded between November 19, 1991 and October 4, 2015 in the EMR of the Faculty of Medicine at the University of Miyazaki Hospital. The target data for our experiments involved medical treatment

```
Algorithm 4: GenClosedSeqs
Input  : $PD$: a projected T-sequential database, $seq$: a
         T-sequence, $MinSup$: a minimum support, $\beta$: range
         parameter
Output : $CS$: noisy-added closed T-sequences with prefix $seq$
begin
    $CS_{seq} \leftarrow \emptyset$;
    $F \leftarrow \{f : (s \subseteq PD, f \in s) \wedge (Sup(f) \geq Size(\text{PD}) \times MinSup)\}$;
    if $(Sup(seq) \geq Size(PD) \times MinSup \times (1-\beta)) \wedge (F \neq \emptyset)$ then
        if $seq.closed() == true$ then
            $CS_{seq} \leftarrow seq$;
        for $f \in F$ do
            $CS_{seq} \leftarrow GenClosedSeqs(PD, seq + f, MinSup)$
              $//seq + f$: append $f$ to $seq$;
```

Table 3: Characteristics of the two datasets

Dataset	CFS	Tur-bt
The number of sequences	271	514
The average length	27.39	86.5
The minimum length	10	11
The maximum length	549	1999

pathways for *Cryptorchidism Fusion Surgery (CFS)* and *Transurethral Resection of a Bladder tumor (Tur-bt)*. We chose these two clinical pathways because CFS is representative of clinical pathways for which the flow of medical treatments is immobilized, whereas Tur-bt is a clinical pathway for which the flow is not well defined. The characteristics of the two datasets are given in Table 3.

Utility Measures: To calculate the utility of the algorithm, we adopted the *F-score* metric, which is widely used for private data analysis [2, 3, 17, 18]. For each *MinSup* setting, we extracted the *Original Results* (*OR*) as the set of the longest frequent sequences having the highest number of items. These sequences were generated without adding noise. We generated the *Noised Results* (*NR*) as the sets of longest frequent sequences generated by the proposed algorithm for various ϵ and β settings. We compared each *Noised Results* set to the *Original Results* by calculating its *F-score*. A higher *F-score* means higher utility. The reason for selecting the longest frequent sequences was because the feedback from medical workers indicated that they would contain all items needed for typical pathways.

Definition 9. ***F-score***
F-score is calculated via recall and precision defined as below. Supposed that $OR = \{s_i | i = 1, ..., n\}$ *and* $NR = \{s'_j | j = 1, ..., m\}$ *then*

$$recall = \frac{\sum_{i=1}^{n} \frac{\sum_{j=1}^{m} LCM(s_i, s'_j)}{s_i.length}}{\#common\_sequences}$$

$$precision = \frac{\sum_{i=1}^{n} \frac{\sum_{j=1}^{m} LCM(s_i, s'_j)}{s_j.length}}{\#common\_sequences}$$

and the F-score is the harmonic mean of recall and precision:

$$F - score = 2 \times \frac{recall \times precision}{recall + precision}$$

Here, $LCM(s_i, s'_j)$ *is the length of the longest common sequences of two sequences* s_i *and* s'_j*, #common_sequences is the total number of longest common sequences in comparison.*

5.2 Experimental Results and Discussion

We conducted several experiments to evaluate the two proposed algorithms T-CSpan-DP and T-CSpan-DPe described in Section 4, and to evaluate the effectiveness of parameters ϵ and β to the performance of the proposed algorithms. We set the value of *MinSup* to 0.1, 0.08, 0.06 and 0.05 for CFS dataset, and to 0.3, 0.25, 0.2, 0.15 for Tur-bt dataset. Here, generally the smaller the *MinSup*, the larger the outputs of the SPM algorithm as more number of sequences can be extracted as the frequent patterns.

5.2.1 *Evaluation of Two Proposed Algorithms.*

Figures 1 and 2 show the *F-score* results of the two proposed algorithms for the two datasets, with variable ϵ budgets. A larger ϵ indicates the bigger gap between the true and the noise frequent values. β was fixed to 0.0.

It is observed that the T-CSpan-DPe gained better *F-score* results than T-CSpan-DP in almost all cases. Especially, T-CSpan-DPe improved at most 32 points, which is corresponding to approximately 50%, for CFS with $MinSup = 0.08$, $\epsilon = 0.01$ (0.95 vs. 0.63). Only apart from Tur-bt dataset with $MinSup = 0.3$ (Figure 2a) that T-CSpan-DPe slightly delivered worse results, however the gap is small, i.e. 3 points.

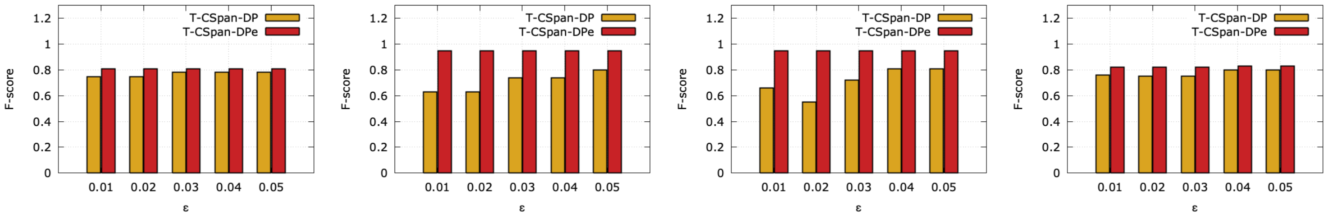
Figures 3 and 4 present the absolute noise value added into the calculation of the two proposed methods for the two datasets. In all cases, there is more noise added in the T-CSpan-DP than in the T-CSpan-DPe. It is understandable that as in the T-CSpan-DP, the number of adding noises is in proportion with the number of frequent items (**Algorithm 1**, line 7 to 9). However, in the T-CSpan-DPe, adding noise occurs only at the final step, i.e. after getting all the frequent closed sequences (**Algorithm 3**, line 11 to 13).

Moreover, when *MinSup* decreases, the amount of noise increases as more sequences can be considered as frequent patterns. In each *MinSup*, increasing ϵ decreases the amount of added noises in both algorithms because the range if noise values becomes narrower.

From the perspective of utility and amount of noise, the T-CSpan-DPe delivered better performance. Even in Tur-bt dataset and $MinSup = 0.3$, *F-score* was slightly worse, however, the amount of added noise was significantly smaller (about 43% of the T-CSpan-DP).

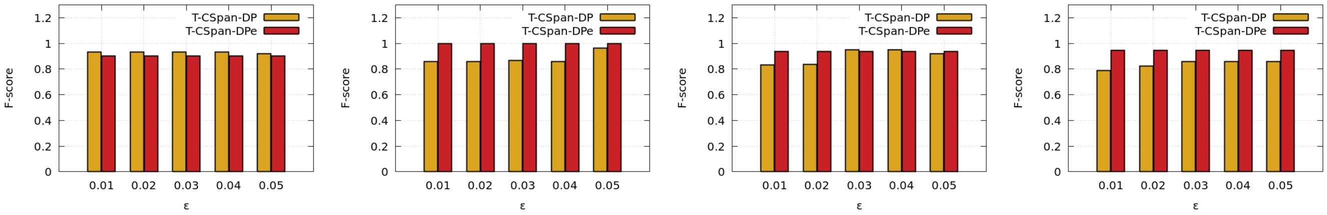
5.2.2 *Evaluation of Parameters.*

As T-CSpan-DPe gained better performance than T-CSpan-DP, in this section, we focus on evaluation of the effectiveness of parameters ϵ and β to the performance of T-CSpan-DPe. We changed the values of β to 0.0, 0.1 and 0.2. The



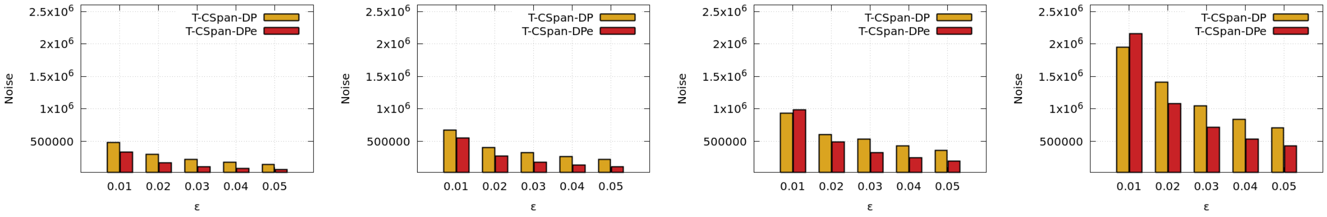
a) MinSup=0.1 **b) MinSup=0.08** **c) MinSup=0.06** **d) MinSup=0.04**

Figure 1: F-score results of the CFS dataset



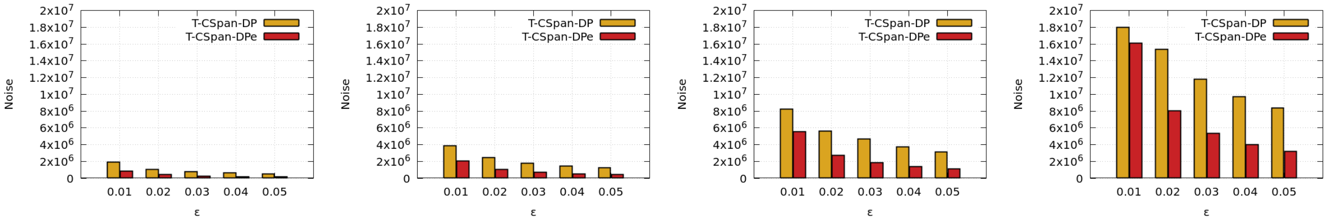
a) MinSup=0.3 **b) MinSup=0.25** **c) MinSup=0.2** **d) MinSup=0.15**

Figure 2: F-score results of the Tur-bt dataset



a) MinSup=0.1 **b) MinSup=0.08** **c) MinSup=0.06** **d) MinSup=0.04**

Figure 3: Amount of added noise of the CFS dataset

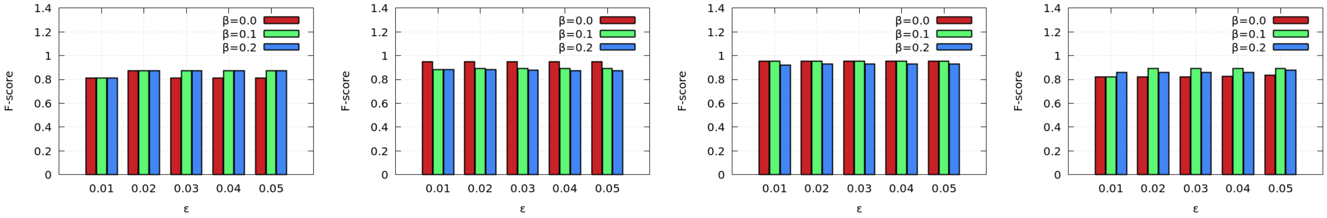


a) MinSup=0.3 **b) MinSup=0.25** **c) MinSup=0.2** **d) MinSup=0.15**

Figure 4: Amount of added noise of the *Tur-Bt* dataset

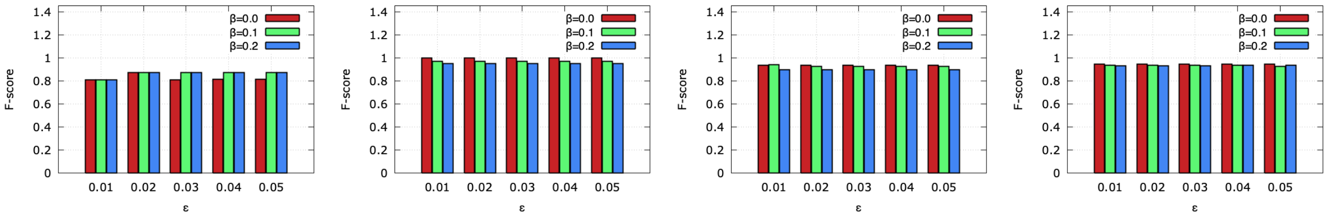
higher the value of β, the more sequences can be considered as frequent patterns.

Figures 5 and 6 show the *F-score* results of T-CSpan-DPe. From these figures, it is observed that T-CSpan-DPe is robust as the results are stable with the variants of ϵ and β. In all settings of ϵ, the *F-score* results are almost the same. Increasing β slightly increases or decreases the results, however, the gap is not considerably large. The biggest difference was 7 points at CFS dataset, $MinSup = 0.08$, $\epsilon = 0.01$ (Figure 5b).



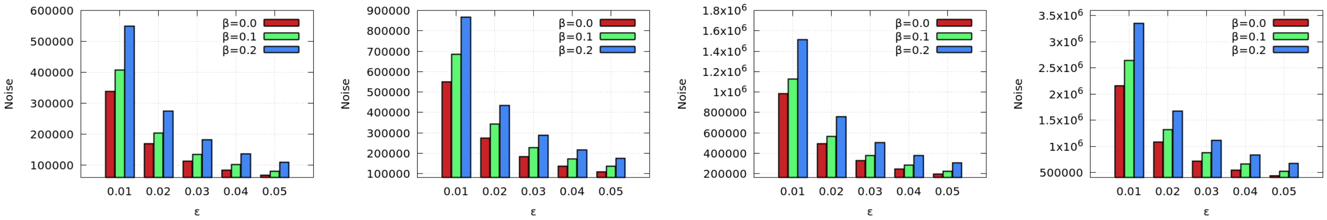
a) MinSup=0.1 **b) MinSup=0.08** **c) MinSup=0.06** **d) MinSup=0.04**

Figure 5: F-score results of T-CSpan-DPe for the CFS dataset



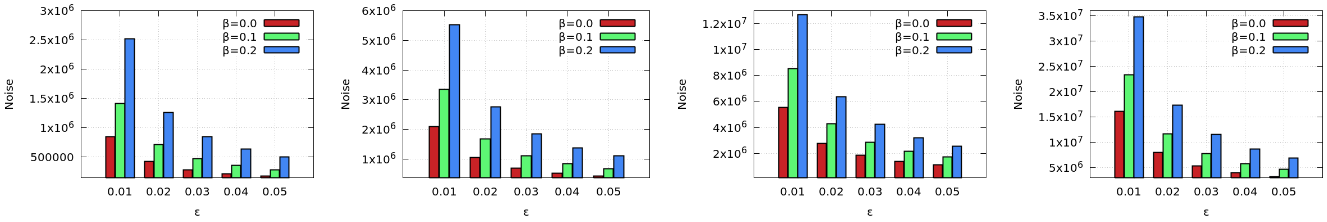
a) MinSup=0.3 **b) MinSup=0.25** **c) MinSup=0.2** **d) MinSup=0.15**

Figure 6: F-score results of T-CSpan-DPe for Tur-bt dataset



a) MinSup=0.1 **b) MinSup=0.08** **c) MinSup=0.06** **d) MinSup=0.04**

Figure 7: Amount of added noise of T-CSpan-DPe for the CFS dataset



a) MinSup=0.3 **b) MinSup=0.25** **c) MinSup=0.2** **d) MinSup=0.15**

Figure 8: Amount of added noise of T-CSpan-DPe for Tur-bt dataset

Figures 7 and 8 present the amount of noise added during calculation of T-CSpan-DPe. It is seen that the amount of added noise is decreasing with the increasing of ϵ, which is similar with the results at Section 5.2.1. However, more noise is added with the larger β as more sequences are counted. As a result, even minimum ϵ and β are able to provide high utility.

5.3 Experimental Evaluation Remarks

We evaluated the two proposed algorithms, i.e. T-CSpanDP and T-CSpanDPe, using real data from EMR system. Although the two algorithms limit adding noise to only a apart of closed sequences, from the experimental results, T-CSpan-DPe was superior to T-CSpan-DP as it gained up to 50% better utility performance with smaller amount of added noise. The reason is that T-CSpan-DPe only adds noise after calculating frequency values of closed sequences, while T-CSpan-DP adds noise during the calculating. The process of adding noise invoked in T-CSpan-DP is proportional to the number of frequent items, which is very large. Moreover, it is also confirmed from the experimental results that T-CSpan-DPe is robust as it is almost not affected by parameters, so it can provide high mining utility with minimum privacy budget.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed two algorithms for SPM considering time interval in EMR systems with satisfying differential privacy. To maintain high utility, noise was added only to the candidate closed sequences. From the perspective of utility and amount of added noise, compared to T-CSpan-DP, T-CSpan-DPe is a better choice because it obtains better utility while the amount of noise is kept smaller as noise is only added at final step deciding which sequences are frequent patterns.

From the experimental results with a real medical dataset, it is verified that T-CSpan-DPe is a robust algorithm as the high utility performance was not affected by parameters. As a result, the proposed algorithm has high potential to be applied in practice that it is able to achieve high utility with minimum privacy budget and noise.

In the future, we would like to evaluate the algorithm with other datasets. Moreover, we also plan to improve the algorithm via well controlling the amount of added noise while maintaining high utility, for example studying mechanisms other than the Laplace mechanism. We also would like to extend our work to other fields than EMR data, such as medical insurance claim data, retailing data, traveler record data.

ACKNOWLEDGEMENT

This research has been supported by the Health Labour Sciences Research Grant (Japanese Ministry of Health, Labour and Welfare) and the Kayamori Foundation of Information Science Advancement.

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1995. Mining Sequential Patterns. In *Proc. the 11th International Conference on Data Engineering (ICDE)*. IEEE, 3–14.

[2] Luca Bonomi and Li Xiong. 2013. A Two-phase Algorithm for Mining Sequential Patterns with Differential Privacy. In *Proc. the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13)*. 269–278.

[3] Rui Chen, Gergely Acs, and Claude Castelluccia. 2012. Differentially Private Sequential Data Publication via Variable-length N-grams. In *Proc. the 2012 ACM Conference on Computer and Communications Security (CCS '12)*. 638–649.

[4] Yen-Liang Chen, Mei-Ching Chiang, and Ming-Tat Ko. 2003. Discovering Time-interval Sequential Patterns in Sequence Databases. *Expert Systems with Applications* 25, 3 (2003), 343–354.

[5] Cynthia Dwork. 2006. Differential Privacy. In *Proc. the 2006 International Colloquium on Automata, Languages and Programming (ICALP '06)*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–12.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 265–284.

[7] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. 2008. Composition Attacks and Auxiliary Information in Data Privacy. In *Proc. the 14th ACM International Conference on Knowledge Discovery and Data Mining (KDD '08)*. 265–273.

[8] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and MC Hsu. 2001. Prefixspan: Mining Sequential Patterns Efficiently by Prefix-projected Pattern Growth. In *Proc. the 17th International Conference on Data Engineering (ICDE)*. 215–224.

[9] Shoji Hirano and Shusaku Tsumoto. 2013. Clustering of Order Sequences based on the Typicalness Index for Finding Clinical Pathway Candidates. In *Proc. the 13th International Conference on Data Mining Workshop (ICDMW)*. IEEE, 206–210.

[10] Hieu Hanh Le, Edman Henrik, Yuichi Honda, Muneo Kushima, Tomoyoshi Yamazaki, Kenji Araki, and Haruo Yokota. 2017. Fast Generation of Clinical Pathways Including Time Intervals in Sequential Pattern Mining on Electronic Medical Record Systems. In *Proc. the 4th International Conference on Computer Science and Computational Intelligent (CSCI)*. 1726–1731.

[11] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *Proc. the 23rd IEEE International Conference on Data Engineering (ICDE'07)*. 106–115.

[12] V Purushothama Raju and GP Saradhi Varma. 2015. Mining Closed Sequential Patterns in Large Sequence Databases. *International Journal of Database Management Systems* 7, 1 (2015), 29–39.

[13] Latanya Sweeney, Akua Abu, and Julia Winn. 2013. Identifying Participants in the Personal Genome Project by Name (A Re-identification Experiment). *CoRR* abs/1304.7605 (2013). arXiv:1304.7605 http://arxiv.org/abs/1304.7605

[14] K. Uragaki, T. Hosaka, Y. Arahori, M. Kushima, T. Yamazaki, K. Araki, and H. Yokota. 2016. Sequential Pattern Mining on Electronic Medical Records with Handling Time Intervals and the Efficacy of Medicines. In *Proc. 2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 20–25.

[15] Bindu Vinay and Traian Marius Truta. 2006. Privacy Protection: p-Sensitive k-Anonymity Property. In *Proc. the 22nd International Conference on Data Engineering Workshops (ICDEW'06)*. 94–103.

[16] Larry Wasserman and Shuheng Zhou. 2010. A Statistical Framework for Differential Privacy. *J. Amer. Statist. Assoc.* 105, 489 (2010), 375–389.

[17] Shengzhi Xu, Xiang Cheng, Sen Su, Ke Xiao, and Li Xiong. 2016. Differentially Private Frequent Sequence Mining. *IEEE Transactional on Knowledge and Data Engineering* 28, 11 (2016), 2910–2926.

[18] Chen Zeng, Jeffrey F. Naughton, and Jin-Yi Cai. 2012. On Differentially Private Frequent Itemset Mining. *PVLDB* 6, 1 (2012), 25–36.

A data mining approach for predicting main-engine rotational speed from vessel-data measurements

Dimitrios Kaklis
Department of Informatics and Telematics, Harokopio University of Athens & NCSR Demokritos & Danaos Shipping Co.
Athens, Greece
dkaklis@iit.demokritos.gr

George Giannakopoulos
Institute of Informatics and Telecommunications, NCSR Demokritos
Athens, Greece
ggianna@iit.demokritos.gr

Iraklis Varlamis
Department of Informatics and Telematics, Harokopio University of Athens
Athens, Greece
varlamis@hua.gr

Constantine D. Spyropoulos
Institute of Informatics and Telecommunications, NCSR Demokritos
Athens, Greece
costass@iit.demokritos.gr

Takis J. Varelas
Danaos Shipping Co.
Piraeus, Greece
drc@danaos.gr

ABSTRACT

In this work we face the challenge of estimating a ship's main-engine rotational speed from vessel data series, in the context of sea vessel route optimization. To this end, we study the value of different vessel data types as predictors of the engine rotational speed. As a result, we utilize speed data under a time-series view and examine how extracting locally-aware prediction models affects the learning performance. We apply two different approaches: the first utilizes clustering as a pre-processing step to the creation of many local models; the second builds upon splines to predict the target value. Given the above, we show that clustering can improve performance and demonstrate how the number of clusters affects the outcome. We also show that splines perform in a promising manner, but do not clearly outperform other methods. On the other hand, we show that spline regression combined with a Delaunay partitioning offers most competitive results.

CCS CONCEPTS

• **Information systems → Clustering**; *Data mining*; • **Computing methodologies → Machine learning**; • **Theory of computation** → *Pattern matching*; *Unsupervised learning and clustering*; • **Applied computing** → *Multi-criterion optimization and decision-making*;

KEYWORDS

time-series forecasting, machine learning, Delaunay triangulation, splines, multivariate regression analysis
Topics Machine learning, Clustering, Time-series analysis

ACM Reference format:
Dimitrios Kaklis, George Giannakopoulos, Iraklis Varlamis, Constantine D. Spyropoulos, and Takis J. Varelas. 2019. A data mining approach for predicting main-engine rotational speed from vessel-data measurements . In *Proceedings of 23rd International Database Engineering and Applications Symposium, Athens, Greece, June 10–12, 2019 (IDEAS'19),* 10 pages.
https://doi.org/10.1145/3331076.3331123

1 INTRODUCTION

Liner shipping companies can benefit significantly by improving ship scheduling and cost analysis in service route planning using computational methods. Furthermore, since there is a strong demand for ships to reduce their emissions, a number of current research activities focus on estimating shipping emissions and developing mitigating solutions to tackle the problem (e.g. [30]). In addition, volatility in fuel prices constitutes a major problem for shipping companies as fuel makes up for 60% of the overall ship operating cost [6]. As a result, modern ship management moves towards energy-efficient procedures and operations, aiming to reduce energy consumption for lowering management costs and thereby maintaining a competitive position in the market while reducing the corresponding environmental impact.

Routing optimization has been a major problem in shipping industry for over three decades and remains one of the research topics of primary interest for the maritime community. Especially nowadays, when new technologies and new concepts, such as *Big Data*, *Data Mining* and *Pattern Recognition* and new methods of data acquisition (AIS data), are overthrowing traditional ways of science exploration, data-driven maritime research is gaining in attention. The automatic identification system (AIS) is an automatic tracking and self-reporting system for identifying and locating vessels by electronically exchanging data among other nearby ships, AIS base

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331123

stations and satellites. The widespread use of AIS allowed vessel tracking and increased the availability of ship trajectory data.

The problem of optimal-route-planning takes into consideration the objectives of ship owners for energy consumption and on-time delivery of goods and the restrictions set by the regulatory framework (national regulations, IMO etc). Regardless the specific constraints, what makes the optimal-route-problem so challenging is the time-varying character of weather conditions during the voyage of the vessel. In this work the optimal-route-problem is mainly examined under the aspect of Fuel Oil Consumption (*FOC*) and an optimal-route is this that minimizes the vessel's FOC for a given destination.

As it is well known from ship-powering literature, *FOC* is closely related with the rotational speed (measured in revolutions per minute - *RPM*) of the main engine. In this connection, the optimal route problem could be significantly simplified if a good predictive model for *RPM* is available. To elaborate further along this path, this article summarizes the current status of our work to couple ship's velocity *V* with main-engine's *RPM* in the context of a non-convex regularized regression estimation problem and in conjunction with the fact that, as marine engineering points out, there is a strong correlation between these two factors. Coupling *V* with *RPM* will give ship operator the benefit of a tool that does not impose installation requirements on the ship, like sensors for gathering data, instead it can be readily used by only getting from satellites the position of the ship, calculating its speed and getting the weather conditions at each time interval. In the same time, it is a first step towards integrating input from more sources (including weather and sea condition data) and allowing the creation of data driven models (black box models) that are able to predict and optimize vessel consumption.

The rest of the paper is organized as follows: In section 2, a summary of the literature related to the routing optimization problem in maritime industry is provided and analyzed. Section 3 presents the motivation behind the work of this paper and summarizes our initial exploratory experiments. Section 4 describes a formulation of the problem at hand and gives an overview of the proposed algorithm. Section 5 depicts and interprets the experimental results, combined with statistical testing. Finally, Section 7 provides the main conclusions of this work and outlines the next steps.

2 RELATED WORK

Following strong regulatory and societal demand for ships to reduce their emissions, current research activities focus on estimating global shipping emissions and develop mitigating solutions to tackle the problem, e.g. [30]. In addition, the increase and volatility in fuel prices constitute a major problem for shipping companies as fuel contributes approximately 60% to the overall ship operating cost [6]. As a result, shipping companies move towards taking on board energy efficient procedures and operations for reducing energy consumption and thereby maintain their competitive position in the market as well as reduce the environmental impact. There is a plethora of theoretical papers related to ship route optimization, starting as early as 1960 [31] and evolving from using simple concepts, such as the so-called isochrone and isopone methods [7], to more elaborate and rigorous approaches, such as optimal control [29], dynamic programming [24], graph theory [25] and evolutionary algorithms [26].

Numerous studies in different disciplines have been undertaken to predict the fuel consumption by using ANN models [27] . ANN is found to be the domain for many successful applications involving prediction tasks, such as modelling and prediction of energy-engineering systems [22], prediction of the energy consumption of passive solar buildings [10], developing energy system and forecast of energy consumption [1], and analysis of emissions reduction [20]. There are also some relevant reports of ANN's being used for implementing decision-support systems in various subjects, such as solving the buffer allocation problem in reliable production [28], developing environmental emergency decision-support systems [24], risk assessment on prediction of terrorism insurgency [11] and modeling of simulation metamodel [2]. ANNs have been used to predict specific fuel consumption and exhaust temperature of a Diesel engine for various injection timings [21].

The optimization objectives in the ship routing problem are usually the minimization of the voyage time, fuel consumption and voyage risk. The approaches, which have appeared so-far in the pertinent literature, can be classified in two large categories: **(a)** *Vessel-based optimization*, where we optimize a given route with respect to vessel characteristics, e.g., vessel speed, main-engine rotational speed, trim, roll, heave and pitch motions, and **(b)** *Condition-based optimization*, where we optimize a given route by taking into account environmental data, e.g., wind (speed, direction), wave (height, frequency, direction), currents, etc. The aforementioned methods utilize techniques that can be separated into three main categories: **(a)** Analytical approaches trying to tackle the problem with the use of exact(NP-complete) and/or heuristic algorithms like label - setting algorithms , non-linear integer programming, simulated annealing [15]; **(b)** Data-oriented approaches that combine vessel-trajectory data, gathered from sensors or satellites (AIS data), with Machine- and Deep-Learning algorithms [23]; **(c)** Approaches where ML (machine learning) methods, e.g., Box Models: White, Black and Grey Box Models (WBM, BBM, GBM), are combined with analytical methods, e.g., the equations of motions of a freely floating body moving with constant forward speed (WBM), in order to increase the accuracy of a regression method in a ML model (BBM) [3].

Finally, methods that refine the voyage grid (map) in areas of critical interest involving, e.g., weather conditions, emission control areas (ECA, SECA: sulfur-oriented ECA's), high-risk zones (piracy), and choose from a set of optimal routes the best in terms of FOC and safety (PARETO optimal solutions, Genetic Algorithms) [12] must also be referenced.

3 MOTIVATION

The motivation for the current work came directly from a business need for the optimisation of the ship engine usage (RPM) in relation to FOC. Based on this requirement, we attempt first to perform an exploratory analysis on a real dataset in order to understand the nature of the FOC - RPM relation. Given the rich and composite feature set of the FOC prediction problem, before training any multi-parameter prediction model it is important to study the effect of each parameter separately. The exploratory analysis was performed on a dataset comprising 10^6 observations from multiple ships and allowed us to determine which feature is most appropriate for the prediction of FOC (Fuel Oil consumption). Initial experiments on the complete dataset were performed using feature selection algorithms in order to rank the features by importance. The Random Forest regression was used for selecting the most informative features. The eight top ranked features on the basis of RF regression are depicted in Table 1.

Feature importance		
Feature Name	Importance	Description
RPM	0.98353	Main engine revolutions per minute.
STW	0.00365	Speed through water.
Speed Overground	0.00266	Speed of the ship with respect to the ground.
Apparent wind speed	0.00133	The relative speed, i.e., the speed experienced by an observer or a measuring instrument on the ship.
Port mid draft	0.00075	Draft amidships on the port side of the ship; port is the left-hand side of a vessel facing forward.
STBD mid draft	0.00042	Draft amidships on the starboard side of the ship; starboard is the right-hand side, facing forward.
Mid draft	0.00075	Draft amidships.
Apparent wind angle	0.0007	The relative angle, i.e., the angle experienced by an observer or a measuring instrument on the ship.

Table 1: The top ranked features by importance, using Random Forest regression.

It is clear from Table 1 that RPM plays a pivotal role in the prediction of FOC. Based on this finding, it seems reasonable to develop a predictive model for FOC using RPM only, since it has maximum importance and is much easier to measure than other features (e.g. wind speed or draft). By measuring the correlation between RPM and each of the remaining seven features of Table 1, using **PPMCC** (Pearson Product Moment Correlation Coefficient), showed an extremely high linear relation (0.92) between RPM and speed overground a result that is also aligned with Figure 1 which confirms a strong linear relationship between the two variables.

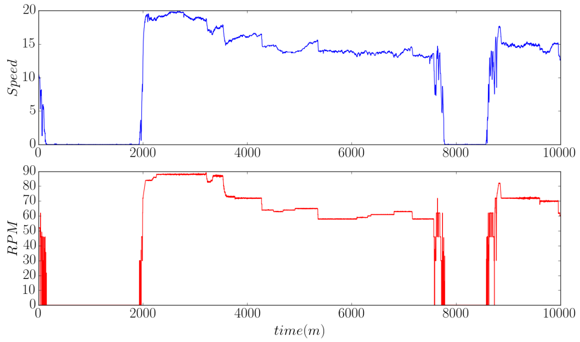


Figure 1: A sample plot of Main-Engine's rotational speed (RPM) and observed speed during a vessel's route (courtesy of DANAOS Shipping Co.)

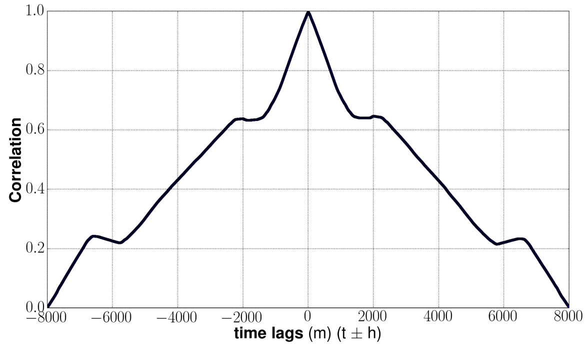


Figure 2: The correlogram of RPM and V during a vessel's route

A survey of the pertinent literature on Naval Architecture and Marine Engineering shows that there is no robust, low complexity, analytical relation between *RPM* and *V*. On the other hand, significant work has been done in complex, time-consuming methods that perform well, while taking into account various related factors, such as geometric and hydrodynamic ones [17]. Thus, our effort of finding a way to efficiently predict *RPM* from *V* utilizing data-driven based methods is well justified. From ship hydrodynamics it is well known that the , where *Q* is the torque absorbed by the propeller of the ship. Then, recalling standard resistance and propulsion theory of ships, we can say that, for a given ship, the torque Q is a function depending exclusively on the ratio V/RPM and, as a result, predicting *RPM* from *V* is a decisive step for predicting power and thus optimising fuel cost.

. This claim is also strengthened by recognizing the commercial potential value of a model like this, as velocity *V* is a feature that can be easily measured –even remotely from a satellite– and does not require further installments (e.g. sensors) on board.

In order to further study what happens before and after velocity changes, we plot the correlation coefficient for each lag variable (observations at previous time steps). This gives a quick idea of which lag variables may be good candidates for building a predictive model and how the relationship between the observations and their historic values changes over time.

The correlogram is a commonly used tool for checking randomness in a data set. In time series analysis, the correlogram, also known as an *autocorrelation plot*, is a plot of the sample autocorrelations r_h versus the time lag *h*. The correlogram of Figure 2 presents the lag number along the x-axis (time axis), with values varying between $-8 * 10^3$ and $8 * 10^3$ minutes and the correlation coefficient value (ranging from 0 to 1) along the y-axis. In random behaviors the auto-correlations should be nearly zero for all time-lag separations. In the opposite case, one or more of the auto-correlations should be significantly different than zero. This is the case of RPM in Figure 2, which reveals a strong correlation between *RPM* and *V* mainly for time steps $t\pm1, \ldots t\pm10^3$ (m) that can be utilized to select appropriate lag variables as extra features to our estimators.

4 PROPOSED METHOD

4.1 Problem formulation

A formal definition of the problem of predicting RPMs based on the monitored velocity (V) over ground can be defined as follows: *Given a vessel's speed for n consecutive periods, find a function* $f(V_1, ..., V_n) : R^n \rightarrow R^1$*, which estimates the engine's RPM at moment* t_{n+1}.

If we assume that the relationship between RPM and V is a partially linear function with non-linear segments over time, then it is possible to describe this specific problem as a *linear mixed-effects model (LMM)* [18]. A mixed model is a statistical model incorporating both fixed and random effects. A random-effects model is a kind of hierarchical linear model, which assumes that the data being analysed are drawn from a hierarchy of different populations, whose differences relate to that hierarchy. These models are useful in a wide variety of applications in physical, biological and social sciences. They are particularly useful in settings where repeated measurements are made on the same statistical units (longitudinal study), or where measurements are made on clusters of related statistical units.

The linear mixed-effects model (LMM) is a great way to model regression algorithms between clustered data and explore the heterogeneity between effects within and between groups of similar values [5]. The connection between RPM and V nicely fits the LMM setting, since in most cases there exists some degree of correlation between the two features which implies a linear dependency. Also, in specific moments very similar V values correspond to different values of RPM inducing a non-linear dependency. Analytically, an LMM can be described as:

$$y = T \cdot d + u + \epsilon, \tag{1}$$

where y is a vector containing the previously observed values of the feature we want to predict, T is a known matrix that relates the observations y to the unknown fixed-effect vector d, u is the unknown covariate vector for random effects and, finally, ϵ is the unknown vectors of random errors. Both u and ϵ share zero-mean normal distributions with $cov(u, \epsilon) = 0$.

A way to combine RPM and V through time (using discrete time slots) is to assume that the following model holds true:

$$\begin{aligned} y_i &= f(t_i) + \epsilon_i, \quad i = 1..., n, \quad t_i \in [0, T], \\ y_i &:= RPM_i, \quad f(t_i) := g(V(t_i)) \end{aligned} \tag{2}$$

where RPM_i is ME rotational speed measured at time t_i, $V(t_i)$ is the ship's speed measured at time t_i and $g(V)$ is the sought-for underlying function that, when composed with the known function $V(t)$, gives, for $t = t_i$, the corresponding RPM_i with error ϵ_i. For continuous time t, the above equation can be written as

$$\begin{aligned} y(t) &= f(t) + \epsilon(t), \quad t \in [0, T], \\ y(t) &:= RPM(t), \quad f(t) := g(V(t)) = (g \circ V)(t), \end{aligned} \tag{3}$$

Since the measurement of V and RPM usually results in many noisy observations a function learned from data can have the form of a smoothing spline that balances between goodness and smoothness of fit. A smoothing spline $\hat{\mathrm{f}}(t), t \in [0, 1]$ in the Sobolev space $\mathcal{H}^{m,2}$, consisting of L^2 functions whose weak derivatives of order up to m belong to L^2 as well, is a solution of the following minimisation problem

$$min_{f \in H^{m,2}} \Big[\frac{1}{n}(y - f)^T W(y - f) + \lambda \int_0^1 (f^m(t))^2 dt\Big], \tag{4}$$

where $y = (y_1, ..., y_n)^T$, $f = (f(t_1), ...f(t_n))^T$ and W is a given positive definite matrix accounting for the correlation between the components of the error vector ϵ. The parameter λ controls the trade-off between fidelity-to-the-data and smoothness of fit and is often referred to as the *smoothing parameter.* In [13, 14] it is shown that the solution of (4) can be expressed as:

$$\hat{f}(t) = \sum_{\nu=1}^{m} d_\nu \phi_\nu(t) + \sum_{i=1}^{n} c_i R^1(t, t_i), \tag{5}$$

where $\phi_\nu(t) = t^{\nu-1}/(\nu - 1)$, $\nu = 1, ..., m$ is a set of polynomials and $R^1(s, t) = \int_0^1 (s-u)_+^{m-1}(t-u)_+^{m-1} du/((m-1)!)^2$ with $x_+ = x$ if $x \geq 0$ and $x_+ = 0$ if $x < 0$ otherwise, is a polynomial spline of degree $2m - 1$, yielding the well-known cubic spline for $m = 2$. Denoting $\hat{T} = \{\phi_\nu(t_i)\}_{i=1,\nu=1}^{n,m}$, $\hat{\Sigma} = \{R^1(t_i, t_j\}_{i=1,j=1}^{n,n}$, one can prove that

$$(\hat{f}(t_1), ..., \hat{f}(t_n))^T = \hat{T}d + \hat{u}, \quad \hat{u} := \hat{\Sigma}c, \tag{6}$$

where $c = (c_1, ...c_n)^T$ and $d = (d_1, ..., d_n)^T$ are solutions to the so-called Henderson's Mixed Model Equation (MME) [8]. Finding the spline estimator for the Linear Mixed-Effects Model (LMM) described in Equation (1), can be done using the Best Linear Unbiased Estimators (BLUE) method [9]. As a result, using spline estimators as a model for revealing the underlying relationship between V and RPM seems to be rational and well grounded.

4.2 Partitioning the input space

In order to take advantage of the ability of **Splines** to fit to **LMMs** problems and adapt on the local (temporal) nature of this correlation between RPM and velocity, we build on this modeling theory. For this purpose, we introduce splines as a way of achieving higher accuracy in RPM prediction based on velocity history and we apply clustering algorithms on the vessel's trajectory data in order to find sub-trajectories that share similar velocity values.

For predicting RPM values from velocity (V) measurements using splines, we suggest to cluster the training space to regions with *similar* velocity patterns. Regions must have similar N previous values of V, on the basis that including velocity at N previous time steps, as an extra feature in the training phase, will lead to a higher accuracy when predicting RPM for time $t = t_{i+1}$ as shown in Section 3. Therefore, each cluster will represent subsets of similar distributions, in terms of standard deviation and mean value, with respect to the history of a value V_i at a given time t_i during a route. All velocity instances V that have similar N previous values are grouped in the same cluster in order to build and train different models that represent different distributions. The training data consists of a 2D vector of the form: $(V(t_i), \bar{V}_N(t_i))^T$ with $\bar{V}_N(t_i) = mean[V(t_{i-N}, ..., V(t_i)]$, and the corresponding $RPM(t)$ value.

At the final stage of the evaluation the problem converts to a problem of classification. Each instance $V(t_i)$ of the test (unseen) dataset must be classified to the most similar $cluster_{k(i)}$. From this point and on we predict the corresponding $RPM(t_i)$ value with the specified $model_{k(i)}$ trained on this particular group of data.

The idea behind the proposed piece-wise regression by clustering is that, as previously stated, the relationship between RPM and V is not linear at all times. As Figures 2 and 1 indicate there exists a strong linear relationship between the dependent (RPM) and independent (V) variables, nevertheless there are parts exhibiting a higher-order (non-linear) correlation. This remark guides us to the choice of building different models, each one corresponding to a different part of relation between our variables (RPM,V), in order to improve the overall accuracy.

Given a sample dataset comprising tuples of the form $\{(x_i, y_i), ..., (x_m, y_m)\}$, where $x_i := V(t_i)$ and $y_i := RPM(t_i)$, we assume that the relation between V and RPM, is described by a polynomial regression model of the form:

$$F(x_i) = b_0 + b_1 x_i + b_2 x_i^2 + \ldots + b_n x_i^n + \epsilon_i,\ i = 1, ...m \\ , or\ in\ matrix\ form:\ \ F = A(x_1, ..., x_m)b + \epsilon, \quad (7)$$

where $b = (b_0, b_1, ..., b_n)^T$ is the parameter vector, A is a Vandermonde matrix, also referred as the design matrix, and ϵ_i is the error vector. The problem of building K regression models in K different clusters $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_K$ can be formalized with the aid of (4.2) as follows:

$$F(x) = \begin{cases} A_1(x_{11}, ..., x_{1m_1})b_1, & x_1 = (x_{11}, ..., x_{1m_1})^T \in \mathcal{X}_1, \\ \vdots & \\ A_k(x_{K1}...x_{Km_K})b_K, & x_K = (x_{K1}, ..., x_{Km_K})^T \in \mathcal{X}_k. \end{cases} \quad (8)$$

When choosing K one must take into account the trade-off between fitting the data and avoiding model complexity and overfitting, which may result in poor generalization on unseen data. This is related to one of the most crucial aspects in function learning, known as the trade-off between bias and variance. The value of K can be chosen through cross-validation, with a possible upper-bound dictated by the maximum tolerable complexity of the estimated model. Clustering the data-set and choosing the optimal K plays a crucial role in piece-wise regression analysis as the results of our experiments (see Section 5) point out.

A draft sketch of the algorithm that performs regression on different sections (clusters) of velocity values to predict RPM follows.

The algorithm begins with the set of (velocity, RPM) pairs D which is clustered into k clusters in a way that optimizes the bias-vs-variance trade-off. Then each instance $V(t_i)$ is classified to the "best" cluster D_b in terms of fitness (using the normalized distance d_{ij} from the centroid C_j of each cluster. The model that has been trained by the cluster that has minimum distance is used to predict the corresponding $RPM(t_i)$ value.

5 EXPERIMENTAL EVALUATION

The aim of the experimental evaluation process is to test the applicability and the performance of the proposed methodology in predicting RPM from previous observations of the velocity (V). More specifically, the questions we examine within this experimental section are the following: **(a)** Does the clustering/partitioning of the input space when combined with models trained separately for each cluster affect the prediction performance? **(b)** Given a dataset containing (RMP, V) observations, is there an optimal number of clusters that maximizes prediction performance? **(c)** How does the number of clusters relate to the expected performance? **(d)** Do spline regression performs better than other established baselines? **(e)** How does the combination of spline regression and clustering of the input space perform?

Algorithm 1 Piecewise regression algorithm with clustered data

```
Require: $D = \{(v_1, r_1), ....., (v_m, r_m)\}$: $v_i = V(t_i)$, $r_1 = RPM(t_i)$
Split D into $k$ clusters $D_1, ..., D_k$
foreach $D_j, j \in [1, k]$ do
    $M_j = train\_regression\_model(D_j)$
end
foreach $V(t_i)$ do
    foreach $D_j, j \in [1, k]$ do
        $C_j = centroid(D_j)$
        $d_{ij} = \frac{1}{1 + \sqrt{\|(V(t_i), \bar{V}_N(t_i))^T - \mathrm{C}_j)\|^2}}$
        $\bar{V}_N(t_i) = mean[V(t_{i-N}, ..., V(t_i)]$
        $D_b = \arg\min(d, D_j)$
    end
end
$RPM(t_i) = M_j(V(t_i))$
```

In order to preserve the statistical independence of our results between different datasets, in all the experiments that follow we apply the two-sample Kolmogorov-Smirnov (K-S) test. The K-S test is a non-parametric test of the equality of continuous (or discontinuous), one-dimensional probability distributions that is used to compare one or more samples with a reference probability distribution. The size of train and test subsets for the experiments presented below is set to approximately $4 * 10^3$ and $3 * 10^3$ observations , respectively.

5.1 Regression methods

Apart from **Spline Regression** (Section 4.1), in our experiments we evaluated three more regression techniques namely Linear Regression, Random Forest Regression and Neural Networks as follows:

Linear regression is a classic regression technique, which models the output variables as linear combinations of the input variables. The regression coefficients of the input variables are usually estimated using least-squares error or least absolute-error approaches and the optimization problem is solved efficiently using either quadratic-programming or linear-programming. In order to accommodate non-linearity, when it exists, polynomial regression is an alternative to linear regression analysis.

Random-Forest regression is an ensemble technique used for classification and regression. It starts with constructing a set of decision trees at training time and then outputs the majority output value (in classification tasks) or the mean output value (in regression tasks) of all individual trees. The randomness principle is either covered by choosing a random subset of features or by choosing a random subset of observations to train each individual tree.

Neural Networks is another popular technique for regression and classification tasks. Using Python's Keras framework we defined a Neural Network with one input and four hidden layers, each one consisting of 10 neurons and one output layer. We used rectified linear unit RelU as the activation function of each layer. RelU is defined as $y(x) = max(0, x)$, and is a function that –in contrast to other activation functions– back-propagates the larger percent of the error on the output to update the neuron weights. A stochastic gradient descent process, the AdaGrad-optimizer of the Keras framework- has been used to find the optimal set of weights for the neural network. Each optimization run for 10 epochs (full training cycles on the training set).

5.2 Clustering methods

The clustering techniques used in our experiments are K-means and a triangulation-based clustering algorithm and are briefly explained in the following. The techniques have been tested Î£n datasets of of size 10^q ($q = 3, 4, 5$).

K-means clustering is a vector quantization method, with origins from the field of signal processing, that is widely used for data clustering. Its main aim is to partition the observations (vectors) into K clusters, so that each observation belongs to the cluster with the nearest centroid (representative vector of the cluster). As a result, the data space is partitioned into Voronoi cells.

Triangulation clustering (DC) [4] first partitions the training space in triangles using a triangulation-based method. Delaunay Triangulation (DT) was used in our experiments due to the fact that is intrinsically related to the Voronoi diagram being actually its dual graph. Another reason for opting in favour of DT among other triangulation techniques is its close connection with the so-called *Delaunay Configurations* that, as stated in [19], is closely related with a multivariate extension of the univariate B-Splines used in this paper for approximation.

By selecting a cut-off value p(a value that is used to determine the neighboring points from the adjacency list of each candidate vector $[\mathbf{V}(t_i), \bar{\mathbf{V}}_N(t_i)]\}$) we can find for each point in the training space its neighboring vertices in the resulting graph. By applying a Depth-First-Search (DFS) algorithm it is possible to find isolated subgraph components recursively as depicted in Figure 4, which shows the resulting clusters for the pointset in Fig. 3.

The basic idea behind clustering with triangulation is that it defines the cluster in a much broader manner, than, e.g., K-means, being able to cluster observations in non-spherical neighborhoods. Also K-means, in its general definition used here, doesn't seem to to detect outliers .In contrast with K-means DT based clustering, as depicted in 4 is able to detect and remove outliers from clusters resulting in more "reliable" clusters. Further research for improving this method has to focus on the search of optimal cut-off value p. Both clustering algorithms showed promising performance especially in conjunction with linear and spline regression, respectively.

5.3 The effect of clustering

Initial experiments were conducted for the previously described clustering methods with constant training size of approximately $3 * 10^3$ instances. Specifically, we utilized the algorithm proposed in Section 4.1 for the aforementioned regression methods, for different values of k (clusters) (for k-means clustering) and different cut-off values P (for the triangulation-based clustering). Indicative results are collected in Figures 5 and 6.

Table 2 summarizes the results of the experimental evaluation on five, statistically independent, samples of size $3 * 10^3$ instances using different combinations of clustering (K-means, Delaunay Triangulation (DT)) and regression (linear LR, splines-based SR, random forests RF and neural networks NN). The table reports the covariance of the input variables and the Mean Average Error (MAE) of the predicted values. Results show that *RF* with K-means and *Splines* with *DT* clustering have the best accuracy. However, the optimal number of clusters varies, depending on the time instance that the training sample was drown and therefore its distribution. The *DT*-based methods perform better with the splines model (*SR*) instead of *LR* and in some cases the overall accuracy achieved by the *SR*/*DT* combination is higher than that achieved by *LR* or *RF* combined with any of the clustering methods. Also, the DT clustering method produces better space partitioning than K-means when Spline regression is going to be used for RPM prediction. The results of our experiments are aligned with the theory that K-means is locally isotropic in contrast to *DT* clustering that is moving in the search space for finding neighboring points by using the weighted edges of the Delaunay Triangulation. On the other hand Neural Networks do not seem to work well after clustering as the results of Table 2 indicate.

Experimental Results				
Algorithm	variance	clusterer	MAE	opt #clusters if ≠ 1
SR	63.892	K-means	1.595	19
SR	66.497	K-means	2.527	6
SR	64.693	K-means	1.880	26
SR	63.892	DC	1.405	19
SR	66.497	DC	1.527	6
LR	63.892	K-means	1.58	19
LR	66.497	—	2.474	1
LR	64.693	K-means	1.761	30
LR	63.892	DC	1.580	19
LR	66.497	—	2.474	1
LR	64.693	—	2.340	1
RF	63.892	K-means	1.550	19
RF	66.497	K-means	2.055	5
RF	64.693	—	21.54	1
RF	63.892	DC	1.550	19
RF	66.497	DC	2.202	5
RF	64.693	—	1.880	1
NN	63.892	—	2.2021	1
NN	66.497	K-means	2.055	5
NN	64.693	—	2.141	1
NN	63.892	DC	12.345	19
NN	66.497	DC	9.234	5
NN	64.693	—	4.412	1

Table 2: Results of the experimental evaluation.

The experimental results in Figures 5 and 6 and Table 2 and further results for five different statistically independent subsets of approximately $4 * 10^3$ observations are summarized in Figure 7 that illustrates the mean difference (in error rate) between clustered and non-clustered data for different regression methods.

Results show that clustering improves the regression algorithms performance especially concerning the first three algorithms (i.e. LR, RF, SR). On the other hand, the NN algorithm has worse performance when combined with clustering, which is also obvious from the last rows of Table 2. Another outcome is that Spline regression (SR) exhibits the largest improvement in terms of prediction error compared to the other three regression methods, when we compare performance between the application in the original and the clustered dataset. This result must be further examined in order to search for a connection between the knots of the spline estimator and the clustered input values. Finally, based on the experimental

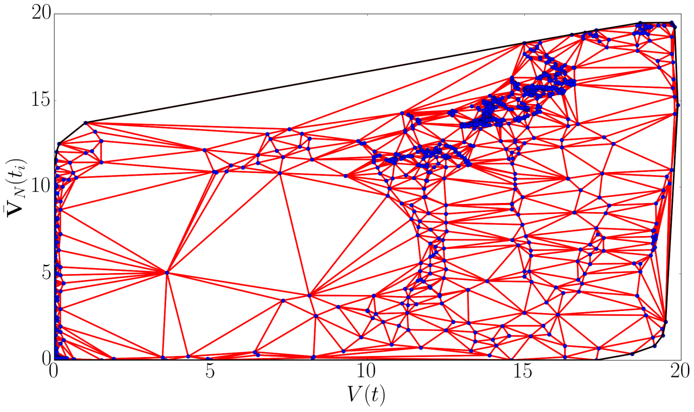


Figure 3: Convex hull and Delaunay Triangulation of a planar training pointset $\{[\mathrm{V}(t_i), \bar{\mathrm{V}}_N(t_i)]\}_{i=1}^{10^3}$

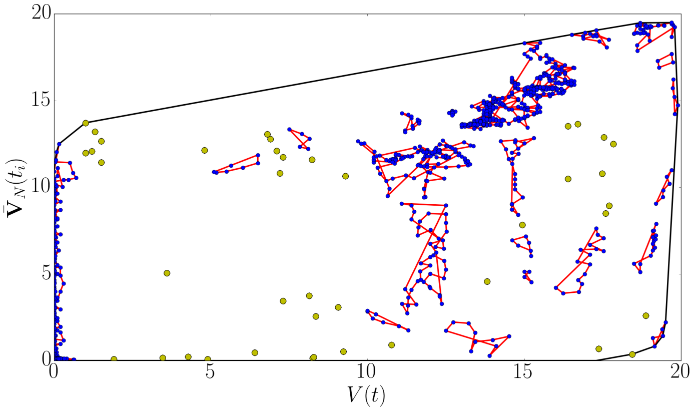


Figure 4: Clustering outcome after applying DFS on the DT of Fig. 3; points belonging to the same cluster are connected with red linear segments.Green points indicate outliers.

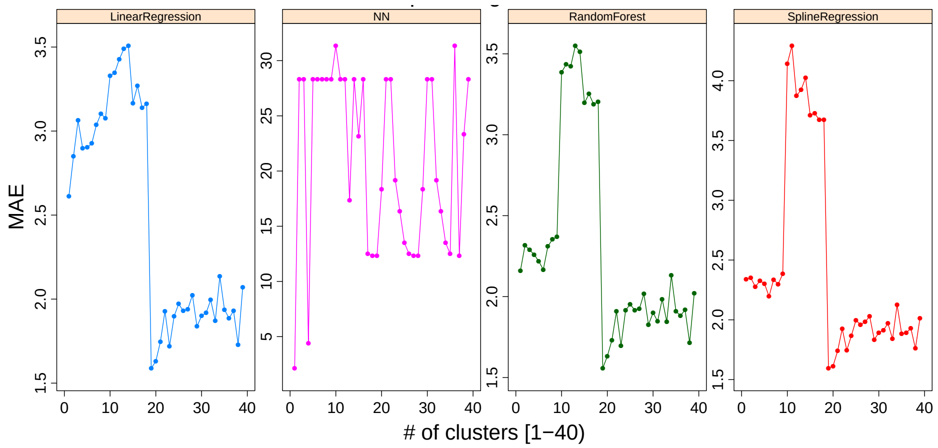


Figure 5: Error convergence with K-means for varying k values

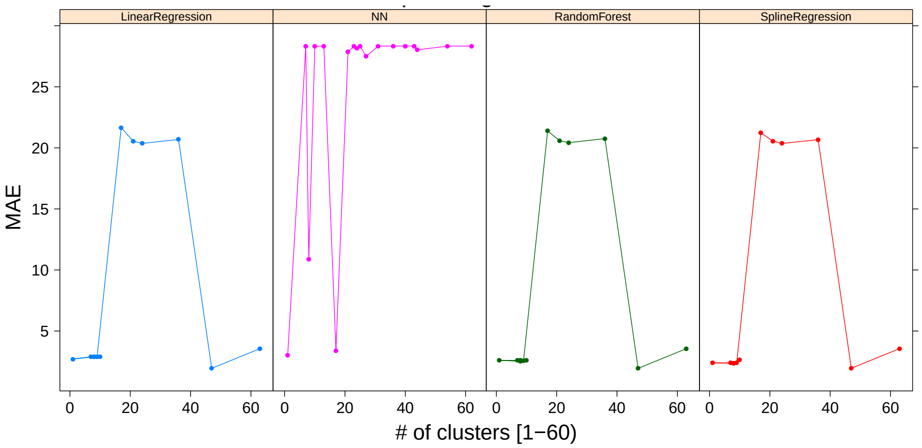


Figure 6: Error convergence with DT clustering for a varying number of clusters.

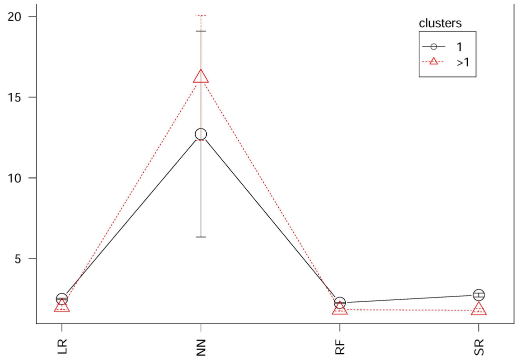


Figure 7: Mean error rate difference of regression models (clustered vs non-clustered samples)

results we can conclude that for all 3 regression algorithms there exists an optimal number of clusters for which they achieve the highest accuracy.

5.4 Finding the optimal number of clusters

The number of clusters of the input variables is proven to be a critical parameter that affects the accuracy of the regression algorithms. In order to statistically prove that the number of clusters plays a significant role in the process of estimating RPM, we perform the Kruskal-Wallis statistical test [16]. This test is a non-parametric approach to the one-way Analysis of Variance (ANOVA) and is used to compare three or more groups on a dependent variable that is measured on at least an ordinal level. The significant result in a Kruskal-Wallis test indicates that there are group differences, but needs a post-hoc procedure to determine which groups are significantly different from each other.

The Kruskal-Wallis test in our case examines the statistical significance between groups of initial parameters, and more specifically between: i) the number of clusters, ii) the clustering method, and iii) the Regression method. The null-hypothesis tested is that there are no significant differences between our groups of features that affect the error rate.

The results indicate that we can accept the null hypothesis for the regression methods (p-value>0.05), while we can reject it for the rest of the groups (clustering method, number of clusters), because their p-value is smaller than the predefined threshold (0.05). As a consequence we can safely claim that the clustering method and the number of clusters have a significant impact on the error rate.

The above results justify the initial idea of applying piece-wise regression on clusters of input values and are indicative of an underlying strong relationship between clustering and regression analysis that must be further examined. They also show that future work must examine many more hyper-parameters and their impact on RPM estimation. For example, the number N of previous time steps involved for building the training vector $(\mathbf{V}(t_i), \bar{\mathbf{V}}_N(t_i))^T$, the variance of the training set, the order and smoothness of the basis functions used in the adopted splines, etc.

5.5 Splines Regression vs other regression methods

The experiments so far showed that clustering of the input space results to higher accuracy for at least 3 out of the 4 proposed regression models. The aim of the next experiment is to test whether the combination of Linear Mixed Models and Spline Regression, as presented in Section 4.1, stands in practice, i.e., test whether, under some constraints, splines perform better than the other two regression methods. As a first step towards this direction, Table 3 presents the results for the optimal number of clusters between 10 statistically independent subsets. These results are associated with the red-dotted line of Figure 7, however, they provide more analytic information.

The results of Table 3 are depicted in the boxplot of Figure 8 below, where the performance of each regression method is compared in terms of absolute value and standard deviation from the median. We easily observe in this plot that Splines (*SR*) and Random Forest (*RF*) perform better than Linear Regression (*LR*) both in terms of accuracy and variance. As far as the comparison between *SR* and

sample	SR	LR	RF
1	1.595/ ***19***	1.588 / ***19***	1.557 /***19***
2	2.027/***6***	2.794 /***2***	2.255 / ***5***
3	2.075 /***2***	2.903/***2***	2.589/***3***
4	1.889 /***26***	1.761/***30***	1.831 / ***39***
5	2.013 / ***31***	2.696/***2***	2.381/***4***
6	2.034 / ***27***	1.5844/***17***	1.667 / ***27***
7	1.4056 / ***23***	2.08 /41	1.574/***23***
8	1.411 /***22***	1.8843 /***52***	1.623/***17***
9	1.436 /***28***	1.588/***27***	1.856 /***44***
10	1.573 /***44***	1.582 /***57***	1.937 /***21***

Table 3: MAE / optimal # of clusters of the three top, in terms of performance, regression methods for the optimal number of clusters being > 1.

RF is concerned, while *SR* appears to perform better than *RF*, the latter exhibits lower variance in error rate than that by Splines.

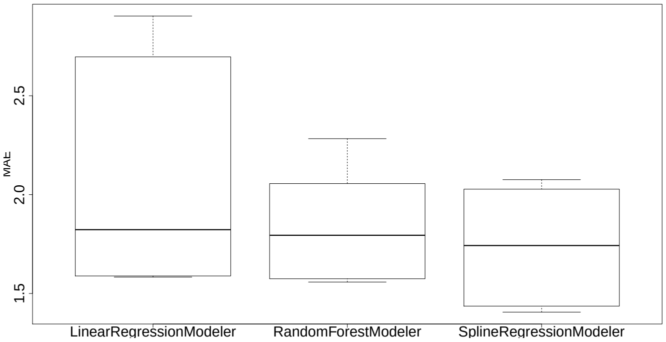


Figure 8: BoxPlot of regression methods compared to error rate

In order to determine which regression method performs better, we apply statistical testing with the results from the Table 3 . Because of our relatively small sample size of 10 samples < 30, we assume non-normality to our dependent variable (the error measured) so we decide to conduct a Wilcoxon signed-rank test, which is the non parametric equivalent of a Paired T-test. Both test are used extensively to compare two groups of dependent (i.e., paired) quantitative data. Wilcoxon can be used in order to determine which algorithm is significantly different than others in terms of accuracy. The null hypothesis tested here is that the true mean error difference between the two regression methods evaluated each time is greater than zero.

The results from the three separate Wilcoxon paired ranked tests indicate that: a) Comparing *RF* with *LR* we get a p-value of $\simeq 0.18 > 0.05$, meaning we can accept the null hypothesis stating that the true mean error difference between the two pairs tested is greater than zero and conclude that *LR* performs better than *RF*. b) Comparing *SR* with *LR* we get a p-value $\simeq 0.03$, which is less than 0.05. Thus, we can reject the null hypothesis. This means that with a confidence level around 95% *SR* performs better than *LR*. c) The same can be stated also for *SR* and *RF*, as the *p*-value $\simeq 0.04$, is close but below the predefined threshold of 0.05 indicating that *SR* performs significantly better than *RF*.

From the statistical tests conducted above, we can conclude that, while it is safe to assume that Splines perform better, with statistical significance, than the other regression methods, the overall performance from the regressors, except NN, combined with clustering was relatively good. In light of these findings one of the next steps of our work would be to study further Spline approximation theory and how clustering affects their accuracy but also to conduct experiments of larger scale with RF and LR.

5.6 Combining splines with clustering

Our experiments so far showed that a strong connection exists between partitioning the input space and the performance of spline regression. Fig. 9 attempts to validate this statement by depicting the mean error difference between the two clustering techniques (K-means and DT-based clustering) for the optimal number of clusters against 5 statistically independent samples consisting of $\approx 4 * 10^2$ observations.

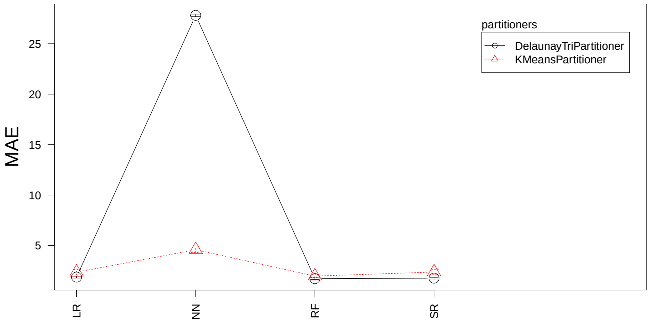


Figure 9: Plot of mean error difference between the two clustering methods for the optimal number of clusters.

More specifically, Figure 9 shows that, while both clustering techniques perform well, DT clustering seems to perform better when it is combined with 3 of the 4 regressors (except from neural networks). Looking at the plot we can also state that spline regression combined with *DT* clustering presented marginally the largest improvement in terms of accuracy compared to the other two regression techniques. This experimental result agrees with pertinent literature, which states a connection between Delaunay partitions and polynomial splines; see Section 4.

6 CONCLUSIONS AND NEXT STEPS

The motivation problem of our work is that of vessel optimal routing by minimizing its FOC (Fuel-Oil Consumption). Via reviewing the pertinent literature on the subject and conducting initial experimentation we concluded that the problem could be handled efficiently if a good predictive model for the *RPM* (revolutionary speed) of the main engine of a vessel moving with known speed *V*were available. Furthermore, access to real industrial data, taken from measurements on-board ships, indicate a strong correlation between *RPM* and *V* on specific time instances during a voyage while others suggested a non-linear relationship between them.

On the basis of the above, we have been led to the idea of developing an *RPM* predictive model that separates the domain in correlated subdomains with respect to velocity V. In this connection, we opted for Spline regression (SR) in order to approximate the underlying function $RPM(V)$ on each subdomain as splines are by their nature continuous piecewise polynomials appropriate for approximation in partitioned domains. The regressor team also included Linear regression (LR), Random Forest (RF) and a baseline Neural Network (NN).

Summarising the results of the approach assembled so far, it seems that spline (SR) and RF (Random-Forrest) regression alongside with partitioning the input space either with K-means or DT-based (Delaunauy Triangulation) algorithm perform better and tend to achieve higher accuracy compared to LR NN. It is also worth-noticing that by enhancing our feature vector with the mean of velocity at N previous time - steps we managed to improve further the accuracy of our predictive scheme.

Besides expanding the scale and variability of our experiments, our short-term future objectives will focus on investigating the effect of several hyper-parameters related to clustering and Spline regression model such as: (i) the optimal cut off value for the DT (triangulation) clustering algorithm and generally the optimal number of clusters for either of the two proposed clustering methods, (ii) the distance metric used for in this paper only Euclidean distance has been tested, (iii) the population and the exact placement of the knots used to approximate the underlying function on each partition and (iv) the order of the spline estimator used to interpolate the data on each partition.

Also further research as far as the tuning process of the hyper-parameter N that control the previous time steps needs to be conducted. Finally, another issue that can be investigated is the practical utilization of the time instance t_i that samples were drawn. A way to achieve this is to include weather conditions at time t_i in our feature space. Wind speed-direction, swell, wave height etc, are some of the parameters that can easily be fed to our existing models or to larger scale - to be built - models like NN that incorporate the existing setting of our proposed algorithm, in order to achieve higher accuracy.

7 ACKNOWLEDGEMENTS

The first author is financially supported by Stavros Niarchos foundation (SNF) . Access to industrial data has been provided by Danaos Shipping Co.

REFERENCES

[1] K. Amirnekooei, M. M. Ardehali, and A. Sadri. 2012. Integrated resource planning for Iran: Development of reference energy system, forecast, and long-term energy-environment plan. *Energy* 46 (2012), 374–385.

[2] B. Can and C. Heavey. 2012. A comparison of genetic programming and artificial neural networks in metamodeling of discrete-event simulation models. *Computers Operations Research* 39 (2012), 424–436.

[3] Andrea Coraddu, Luca Oneto, Francesco Baldi, and Davide Anguita. 2017. Vessels fuel consumption forecast and trim optimisation: a data analytics perspective. *Ocean Engineering* 130 (2017), 351–370.

[4] C Eldershaw and Markus Hegland. 1997. Cluster analysis using triangulation. *Computational Techniques and Applications* (1997), 201–208.

[5] John Fox. 2015. *Applied regression analysis and generalized linear models.* Sage Publications.

[6] Mihalis M Golias, Georgios K Saharidis, Maria Boile, Sotirios Theofanis, and Marianthi G Ierapetritou. 2009. The berth allocation problem: Optimizing vessel arrival time. *Maritime Economics & Logistics* 11, 4 (2009), 358–377.

[7] H Hagiwara and JA Spaans. 1987. Practical weather routing of sail-assisted motor vessels. *The Journal of Navigation* 40, 1 (1987), 96–119.

[8] CR Henderson. 1974. General flexibility of linear model techniques for sire evaluation. *Journal of Dairy Science* 57, 8 (1974), 963–972.

[9] Charles R Henderson. 1975. Best linear unbiased estimation and prediction under a selection model. *Biometrics* (1975), 423–447.

[10] S. A. Kalogirou and M. Bojic. 2000. Artificial neural-networks for the prediction of the energy consumption of a passive solar-building. *Energy* 25 (2000), 479–91.

[11] Neungrit P. Kengpol A. 2014. A decision support methodology with risk assessment on prediction of terrorism insurgency distribution range radius and elapsing time: An empirical case study in Thailand. *Computers Industrial Engineering* (2014), 55–67.

[12] Boram Kim and Tae-Wan Kim. 2017. Weather routing for offshore transportation using genetic algorithm. *Applied Ocean Research* 63 (2017), 262–275.

[13] G.S. Kimeldorf and G. Wahba. 1971. Some Results on Tchebycheffian Spline Functions. *J. Math. Anal. Appl.* 33 (1971), 82–94.

[14] George S Kimeldorf and Grace Wahba. 1970. Spline functions and stochastic processes. *Sankhyā: The Indian Journal of Statistics, Series A* (1970), 173–180.

[15] Odysseas Kosmas and Dimitrios Vlachos. 2012. Simulated annealing for optimal ship routing. *Computers & Operations Research* 39, 3 (2012), 576–581.

[16] William H Kruskal and W Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association* 47, 260 (1952), 583–621.

[17] Edward V Lewis. 1988. Principles of naval architecture, Second revision. *Jersey: SNAME* 2 (1988).

[18] P McCullagh and J. A. Nelder. 1989. *Generalized Linear Models.* Vol. 2nd ed. Chapman and Hall/CRC Press.

[19] Marian Neamtu. 2007. Delaunay configurations and multivariate splines: A generalization of a result of BN Delaunay. *Trans. Amer. Math. Soc.* 359, 7 (2007), 2993–3004.

[20] B. Ozer and E. Gorgun. 2013. ÎŹncecik, S., The scenario analysis on CO2 emission mitigation potential in the Turkish electricity sector: 2006–2030. *Energy* 49 (2013), 395–403.

[21] Adnan Parlak, Yasar Islamoglu, Halit Yasar, and Aysun Egrisogut. 2006. Application of artificial neural network to predict specific fuel consumption and exhaust temperature for a diesel engine. *Applied Thermal Engineering* 26, 8-9 (2006), 824–828.

[22] Kalogirou Sa. 2000. Applications of artificial neural-networks for energy systems. *Applied Energy* 67 (2000), 17–35.

[23] R Savitha, Abdullah Al Mamun, et al. 2017. Regional ocean wave height prediction using sequential learning neural networks. *Ocean Engineering* 129 (2017), 605–612.

[24] Wei Shao, Peilin Zhou, and Sew Kait Thong. 2012. Development of a novel forward dynamic programming method for weather routing. *Journal of marine science and technology* 17, 2 (2012), 239–251.

[25] George Stergiopoulos, Evangelos Valvis, Dimitris Mitrodimas, Dimitrios Lekkas, and Dimitris Gritzalis. 2018. Analyzing Congestion Interdependencies of Ports and Container Ship Routes in the Maritime Network Infrastructure. *IEEE Access* 6 (2018), 63823–63832.

[26] Rafal Szlapczynski and Joanna Szlapczynska. 2012. On evolutionary computing in multi-ship trajectory planning. *Applied Intelligence* 37, 2 (2012), 155–174.

[27] N. K. Togun and S. Baysec. 2010. Prediction of torque and specific fuel consumption of a gasoline engine by using artificial neural Networks. *Applied Energy* 87 (2010), 349–355.

[28] Athanasios K Tsadiras, CT Papadopoulos, and Michael EJ OâĂŹKelly. 2013. An artificial neural network based decision support system for solving the buffer allocation problem in reliable production lines. *Computers & industrial engineering* 66, 4 (2013), 1150–1162.

[29] DS Vlachos. 2004. Optimal ship routing based on wind and wave forecasts. *Applied Numerical Analysis and Computational Mathematics* 1, 2 (2004), 547–551.

[30] Conor Walsh and Alice Bows. 2012. Size matters: exploring the importance of vessel characteristics to inform estimates of shipping emissions. *Applied Energy* 98 (2012), 128–137.

[31] Laura Walther, Anisa Rizvanolli, Mareike Wendebourg, and Carlos Jahn. 2016. Modeling and optimization algorithms in ship weather routing. *International Journal of e-Navigation and Maritime Economy* 4 (2016), 31–45.

Studying forward looking bubbles in Bitcoin/USD exchange rates

Stefano Bistarelli
stefano.bistarelli@unipg.it
University of Perugia

Gianna Figá Talamanca
gianna.figatalamanca@unipg.it
University of Perugia

Francesco Lucarini
francesco.lucarini@studenti.unipg.it
University of Perugia

Ivan Mercanti
ivan.mercanti@imtlucca.it
IMT School for Advanced Studies, Lucca

Abstract

Although Bitcoin is a relatively new subject in Economics, contributions in this topic are growing very fast. Several papers evidenced a bubble behaviour in exchange rates between Bitcoin and traditional currencies. In this paper we explore and give validation to such conjecture, proving also that the bubble effect is due to confidence in Bitcoin future values. This means that Bitcoin price/exchange rate is influenced both by future and past events, but that the bubble behaviour is strictly connected to trust on the future of the Bitcoin system.

CCS Concepts • **Applied computing** → **Digital cash**; • **Information systems** → *Record storage systems*; • **Networks** → Network performance evaluation.

Keywords Bitcoin, Causal-Noncausal Autoregressive models

ACM Reference Format:
Stefano Bistarelli, Gianna Figá Talamanca, Francesco Lucarini, and Ivan Mercanti. 2019. Studying forward looking bubbles in Bitcoin/USD exchange rates. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3331076.3331106

1 Introduction

Bitcoin is a relatively new subject in Economics and Finance, however, such digital currency is fostering a lot of studies, and contributions in this topic are growing very fast. Some of the studies go in the direction of understanding the reasons of special activities in the market. In particular, several papers evidenced a bubble behaviour in exchange rates between Bitcoin (BTC henceforth) and traditional currencies (Euro or Dollars usually) [9, 17]. The aim of this paper is to explore the conjecture that the bubble effect is due to confidence in Bitcoin future values so that its price/exchange rate is influenced both by past events and by views about future ones.

Traditional econometrics models within the class of AutoRegressive Integrated Moving Average (ARIMA) are *backward looking* since the only time-dependence admitted regards the past [6] and are usually referred to as causal models. Recently, models known as Mixed causal-noncausal AutoRegressive (MAR) have been introduced in order to extend time dependence to the future [7, 10, 18] thus reflecting a *backward-forward looking* behaviour.

The paper by Gouriéroux & Hencic [15] represents a valid anchor to refer to, at least in this area of study, as it undertakes a non-causal analysis of the BTC/USD rates in order to predict its future evolution. The present study shares with [15] both the same decomposition of the BTC/USD price in a bubble and in a fundamental part, and the observed time series; though, here the main objective is to investigate whether confidence in future values of the BTC/USD rate (i.e. the *forward looking* part) is the one responsible of the bubble effect, while in [15] the focus was on forecasting future rates. If this is the case, a significant change in the estimated parameters should be detected when the MAR model is estimated separately in the observed time series and in the bubble component. In particular the *forward looking* parameters should be stronger in the bubble part than in the observed price.

The rest of the paper is structured as follows: the firt part is devoted to the economic explanation of our conjecture about the relation of the speculative bubble in BTC/USD exchange rates with the monetary policy of the Bitcoin system; then, in Section 3 the theory behind the *Mixed Causal–Noncausal autoregressive models* is briefly described. Section 4 describes the dataset and Section 5 summarizes the results of the estimation of the MAR model on the observed data. Section 6 gives conclusions and final remarks.

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331106

2 The speculative bubble in BTC/USD rates

By simply watching the trajectory of the BTC/USD exchange rate time series it's easy to notice how often its pattern surges and bursts rapidly mimicking the one of speculative bubbles. The definition of speculative bubble considered in this paper is the one proposed by Blanchard [5] in the framework of rational expectations models where it is assumed that the economic variable of interest, say x_t, has two components: the first one depicts the fundamental path of x_t, while the second represents the bubble effect. In this context a bubble results from the departure of x_t from it's *fundamental path*. In Fig. 1 one of the major bubbles occurred in 2013 for the BTC/USD rate is recorded.

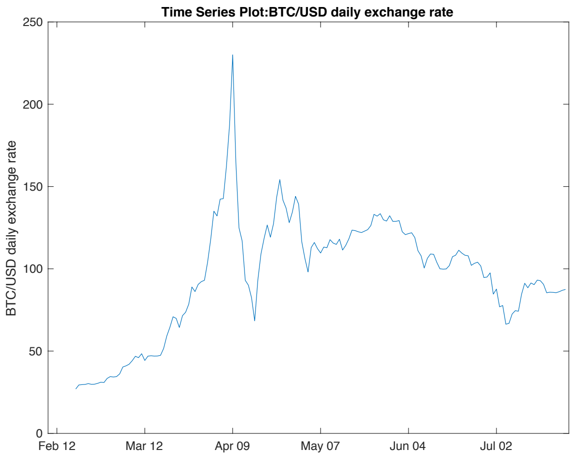


Figure 1. Bitcoin/USD observed time series

Bitcoins are produced through a "mining" process which involves computers (nodes from now on) solving complex mathematical problems (cryptography) to keep the system secure; when the node find a solution to the problem it is rewarded with an amount of Bitcoins which is referred to as "Block reward". The protocol running Bitcoin is programmed to halve every 4 years the "Block reward" by suitably increasing the difficulty of the mathematical problems to be solved. Hence, the volume of new coins will decay to zero with time and the long-term monetary supply will be fixed.

This paper aims at investigating whether the peculiar "deflationary" mechanism running the system's monetary issuance is the main responsible of the formation, and the subsequent crash, of speculative bubbles (against the USD and other currencies). Indeed economic agents, before undertaking any action within the system, already include the system's monetary issuance in their preferences/expectations, *i.e.* they already know that monetary issuance will never diverge from their expectations inasmuch the system has a unalterable monetary policy programmed to ever decrease the monetary issue over time. Therefore as the system grows (think of it as the Gross Domestic Product of a national economy) the demand of Bitcoins will increase, boosting upwards its price against other traditional currencies, given the *ex-ante* fixed monetary supply.

The reason why this issuance mechanism is hereby defined "deflationary" is that as long as the general belief that the system as a whole will keep growing stands, the price against other currencies will inevitably increase, increasing the inter-temporal opportunity cost of spending any given amount of BTC. As a matter of fact since the agents know that the price will increase then they are encouraged to withhold any transaction in BTC and increase their *savings* in BTC. A very interesting effect of such mechanism is that any steep fall in the price may boost the awareness of BTC as a system, potentially increasing it's diffusion among the general public, thus incrementing the aforementioned self–sustaining dynamic [17].

It must be noticed that if the system had a flexible monetary policy, where changes are not known *ex-ante*, then the economic agents within and without the system wouldn't be able to include it in their preferences, thus neutralizing the aforementioned self–sustaining mechanism, even if the bitcoin system is flourishing. After the explanation given above it must be clear now the reason why it is expected and tested below in this study that the speculative bubble in BTC/USD rates is a *forward–looking* phenomena.

3 Mixed causal-noncausal autoregressive models

For a long time, as mentioned by Gouriéroux & Hencic [15], speculative bubbles were considered as nonstationary phenomena and treated similarly to the explosive, stochastic trends due to unit roots. Gouriéroux & Zakoian [11] propose a different approach and assume that the bubbles are rather short-lived explosive patterns caused by extreme valued shocks in a noncausal, *stationary* process. In particular they assume a noncausal AR(1) (Auto Regressive) model, strictly *forward looking*, with Cauchy distributed errors.

A useful feature of such models is that shocks are nonfundamental, combining this trait with the extended time dependance (to the future) allows these models to perfectly fit the peculiar pattern of the aforementioned (definition of) speculative bubble.

3.1 Introduction to noncausality

Let y_t be the observed time series onto which estimate the traditional autoregressive model:

$$\begin{aligned} a(L)\, y_t &= \varepsilon_t \\ (1 - a_1 L - \cdots - a_p L^{-p})\, y_t &= \varepsilon_t \end{aligned} \tag{1}$$

with L being the backshift operator, *i.e.*, $L\, y_t = y_{t-1}$ gives lags and $L^{-1}\, y_t = y_{t+1}$ produces leads and a the autoregressive parameters. It is known [18] that if s out of p of the polynomial's $(a(L))$ *zeros* are inside the unit circle, then the model is *non stationary* causing the impossibility to estimate the traditional autoregressive model. ε_t represent the usual error term of the model.

In Lanne & Saikkonen [18] it is shown that when $p = r + s$, with r being the zeros outside the unit circle, one can factor the polynomial $a\,(z)$ as[1]:

$$a(z) = \varphi^*(z)\,\phi(z) \tag{2}$$

where $\phi(z)$ is the usual causal polynomial of the autoregressive parameters and $\varphi^*(z)$ has its zeros inside the unit circle.

The polynomial $\varphi^*(z)$ can be expressed as:

$$\begin{aligned}\varphi^*(z) &= 1 - \varphi_1^* z - \cdots - \varphi_s^* z^s \\ &= -\varphi_s^* z^s \left(1 + \frac{\varphi_{s-1}^*}{\varphi_s^*}\, z^{-1} + \cdots + \frac{\varphi_1^*}{\varphi_s^*}\, z^{1-s} - \frac{1}{\varphi_s^*}\, z^{-s}\right) \\ &= -\varphi_s^*\, z^s\, \varphi(z^{-1})\end{aligned} \tag{3}$$

where $\varphi(z^{-1}) = 1 - \varphi_1\, z - \cdots - \varphi_s\, z^s$ in view of the fact that $\varphi_{s-j}^* / \varphi_s^* = -\varphi_j$ for $j = 1, \ldots, s$ and $1/\varphi_s^* = \varphi_s$.

Because the zeros of $\varphi^*(z)$ lie inside the unit circle those of $\varphi(z)$ lie outside of the unit circle. Thus, (1) can be written as:

$$\phi(L)\,[-\varphi_s^*\, L^s\, \varphi(L^{-1})] = \varepsilon_t$$

given the decomposition shown in (3). Also, the latter expression can be rearranged as:

$$\phi(L)\,\varphi(L^{-1})\,y_t = \epsilon_t \tag{4}$$

where $\epsilon_t = -(1/\varphi_s^*)\,L^{-s}\varepsilon_t = -(1/\varphi_s^*)\,\varepsilon_{t+s}$. It is important to notice that $E_t[\epsilon_t] \neq \epsilon_t$ since this variable is not determined by any informations available at time point t (see above).

3.2 Mixed Causal-Noncausal Autoregressive Model

The univariate mixed causal-noncausal autoregressive model, denoted MAR(r, s), shown with equation 4 is usually written as:

$$(1 - \phi_1\, L - \cdots - \phi_r\, L^r)(1 - \varphi_1\, L^{-1} - \cdots - \varphi_s\, L^{-s})\, y_t = \epsilon_t \tag{5}$$

When $\varphi_1 = \cdots = \varphi_s = 0$, the process y_t represent a purely causal autoregressive process denoted AR$(r, 0)$:

$$(1 - \phi_1\, L - \cdots - \phi_r\, L^r)\, y_t = \epsilon_t \tag{6}$$

where y_t is regressed on past values, giving the process y_t a *backward looking* autoregressive dynamic.

The process y_t is *purely noncausal* when $\phi_1 = \cdots = \phi_r = 0$, hence defined as:

$$(1 - \varphi_1\, L^{-1} - \cdots - \varphi_s\, L^{-s})\, y_t = \epsilon_t. \tag{7}$$

usually referred to as *forward looking* AR$(0, s)$ process, being the exact counterpart of the model specification given in (6), since it's regressed on future values rather than past ones.

Models containing both lags and leads of the dependent variable are called *mixed causal–noncausal models.*

Assuming that the roots of the causal and noncausal polynomial are outside the unit circle, that is:

$$\phi(z) = 0 \quad \text{per} \quad |z| > 1 \quad \text{e} \quad \varphi(z) = 0 \quad \text{per} \quad |z| > 1 \tag{8}$$

than these conditions imply that the series y_t admits a two-sided Moving Average (MA) representation:

$$y_t = \sum_{j=-\infty}^{\infty} \psi_j\, \epsilon_{t-j} \tag{9}$$

such that $\varphi_j = 0$ for all $j < 0$ implies a purely causal process t and a purely noncausal model when $\varphi_j = 0$ for all $j > 0$ [19]. More in detail, the ψ_jâĂŹs are the coefficients of an infinite order polynomial in positive and negative powers of the Lag operator and such that $\Xi(z) = \sum_{j=-\infty}^{\infty} \psi_j\, z^j = [\Psi(z^{-1})]^{-1}\,[\Phi(z)]^{-1}$.

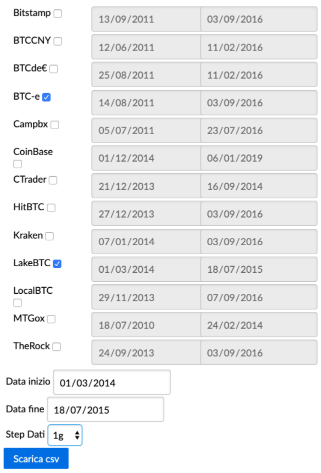
Error terms ϵ_t are assumed *iid* non-Gaussian with $E(|\varepsilon_t|^{\delta}) < \infty\ \forall\ \delta \in (0, 1)$ [11]. Following Gouriéroux & Jasiak [10] the unobserved causal and noncausal components of the process y_t are defined as follows:

$$u_t \equiv \phi(L)\,y_t \;\leftrightarrow\; \varphi(L^{-1})u_t = \epsilon_t, \tag{10}$$

$$v_t \equiv \varphi(L^{-1})\,y_t \leftrightarrow \phi(L)\,v_t = \epsilon_t \tag{11}$$

The specification of these values will prove useful for the following part regarding the estimation of mixed causal-noncausal processes.

The non-Gaussianity assumption for the error term ensures the identifiability of the causal and the noncausal part. Most papers by Lanne & Saikkonen et al. use Student's t_ν distributions, with $\nu \geq 2$ while Gouriéroux et al. rely on the Cauchy or a mixture of Cauchy and Normal distributions. As shown by Hecq et al. [14] it emerges that the Cauchy has too strong fat tails features and many series would have a degree of freedom between 1.5 and 2.5[2].



Bitstamp □	13/09/2011	03/09/2016
BTCCNY □	12/06/2011	11/02/2016
BTCde€ □	25/08/2011	11/02/2016
BTC-e ☑	14/08/2011	03/09/2016
Campbx □	05/07/2011	23/07/2016
CoinBase □	01/12/2014	06/01/2019
CTrader □	21/12/2013	16/09/2014
HitBTC □	27/12/2013	03/09/2016
Kraken □	07/01/2014	03/09/2016
LakeBTC ☑	01/03/2014	18/07/2015
LocalBTC □	29/11/2013	07/09/2016
MTGox □	18/07/2010	24/02/2014
TheRock □	24/09/2013	03/09/2016

Data inizio 01/03/2014

Data fine 18/07/2015

Step Dati 1g

Figure 2. TradeBitcoin data price download panel

[1] In order to maintain the same notation as in Lanne & Saikkonen [18] the polynomial $a(L)$ will be referred to as $a(z)$ for the following proof.

[2] Notice that when $\nu < 2$ then the Student's t expected value is undefined.

4 The Data

The sample consists in 151 observation of the BTC/USD price spanning from February 20 to July 20 2013. The dynamic of the data is shown in Fig.1, where it is possible to notice the speculative bubble behaviour of the BTC/USD path, boosting and bursting rapidly around the month of April. In fact, in the April 2013 there was a famous bubble, commonly called simply the April bubble, that was a rally, all-time high and subsequent crash of the bitcoin exchange rate. The bubble resulted in a momentary all-time high of $266 USD per bitcoin on Mt. Gox[3] on 10th April 2013. Then Mt. Gox suspended trading on 11th April 2013 until 12th April 2013 2 am UTC for a "market cooldown". The value of a single bitcoin fell to a low of $55.59 after the resumption of trading before stabilizing above $100[4] (a price decline of 61%).

The data is obtained from our application, TradeBitcoin [1], part of the suite BlockChainVis[5] [2–4] used for Bitcoin analysis and visualization, is based on finding the price options on the Bitcoin exchange and writing possible arbitrage operations on a database to see if it is possible to correctly perform an arbitrage on the Bitcoin market. It also collects all this data price from 17 different exchanges and it allows to download that data with a detection time of 1 day or 1 hour or 15 minutes (Fig. 2).

4.1 Price decomposition

As a first issue it is important to disentangle the fundamental component from the bubble component of the BTC/USD prices. The fundamental value of the Bitcoin is still under debate. While in [8] it is argued that this fundamental value is zero, in [9] it is linked to the reputation of the Bitcoin system measured by internet queries, moreover it is suggested (still in [9]) that the production cost of Bitcoin, due to the mining process, should be considered as the lower limit of the fundamental value of Bitcoin. Since this study assumes that Bitcoin has a fundamental value indeed, the price will be firstly decomposed following the approach in [15], where the fundamental path of the BTC/USD rate is assumed to be a nonlinear deterministic trend modelled as a 3rd degree polynomial in time and the bubble part is obtained by subtraction from the observed prices, as it is done still in [15]. The other decomposition that will be undertaken builds upon the suggestion made in [9], by setting apart the production cost of Bitcoin and the bubble component using the cost of production model shown in [13].

4.1.1 Nonlinear deterministic trend

As mentioned above the BTC/USD rate is defined as follows:

$$rate_t = trend_t + y_t, \tag{12}$$

with $rate_t$ being the observed prices, $trend_t$ the fundamental component and y_t the bubble component and the estimated trend is given by:

$$trend_t = 0.000073\, t^3 - 0.0316\, t^2 + 3.6590\, t - 3.2951.$$

The corresponding time-series are plotted in Fig. 3.

4.1.2 Production cost as the lower limit of the fundamental value

The Bitcoin production cost model shown in [13] assumes the perspective of a generic *miner* that is deciding whether to mine or not for Bitcoin. The *miner* will decide to join the mining process in case of positive profit expectations and to abandon it on the contrary case. The variables considered to be influencing the mining process and hence the production cost in [13] are: the *block reward* β, the *hashing power* (computational power) of the mining hardware equipment ρ, the *difficulty* set by the network δ, the *cost* per kilowatt-hour[6] $\$ kW/h$ and the *average energy efficiency* $W\,GH/s$ of the mining hardware deployed.

As shown in [13] the expected number of cryptocurrency coins to be mined per day on average given the difficulty and block reward (number of coins issued per successful mining attempt) per unit of hashing power is given by:

$$BTC/day = \frac{\beta\, \rho\, sec_{hr}}{\delta\, 2^{32}}\, hr_{day}$$

sec_{hr}= 3600 being the seconds in 1 hour and hr_{day}= 24 being the hours in a day.

The cost of mining can be expressed as:

$$E_{day} = (\rho/1000)\,(\$\, kW/h \;\; W\,GH/s \;\; hr_{day})$$

with $\$ \; kW/h$ being the electricity cost and $W \; GH/s$ the average energy efficiency. Bitcoin production cost estimates over the considered time span (Feb.-Jul. 2013) are shown in Fig. 4, where it is assumed an average energy efficiency of $W \; GH/s = 500$ as suggested by Garcia et al. [9], a computational power[7] of $GH/s = 1000$, an average global electricity cost of $\$\, kW/h = 0.115$. In 2013 the block reward set by the network was $\beta = 25 \; BTC$, the values of the ever changing difficulty over the considered time-period can be found in the public database https://blockchain.info.

Assuming the lower limit of the fundamental value, given by the aforementioned definition of production cost, as the actual fundamental value, therefore the BTC/USD rate is

[3] https://en.bitcoin.it/wiki/Mt._Gox.

[4] https://bitcoincharts.com/charts/mtgoxUSD#rg5zczsg2013-04-10zeg2013-04-12ztgSzm1g10zm2g25zl.

[5] http://normandy.dmi.unipg.it/blockchainvis/.

[6] In this study we consider the same average cost for electricity that is considered in [13], although it must be noticed that it changes depending on the geographical location of the miner and therefore on the national electricity supplier.

[7] It must be noticed that in this example varying the computational power does not change the cost, *i.e.*, the cost of 1 BTC in USD is only affected by the difficulty, the electricity cost and the average energy efficiency of the mining hardware, increasing the scale of production in this case doesn't lead to economies of scale.

defined similarly as in 12:

$$rate_t = cost_t + yc_t, \qquad (13)$$

with $rate_t$ being the observed prices, yc_t the bubble component and where the fundamental component in this case is given by the aforementioned production cost $cost_t$. Assuming that the production cost is correctly estimated, it must be noticed that the bubble component could be considered as the *added market value.*

The corresponding time series are plotted in Fig. 4

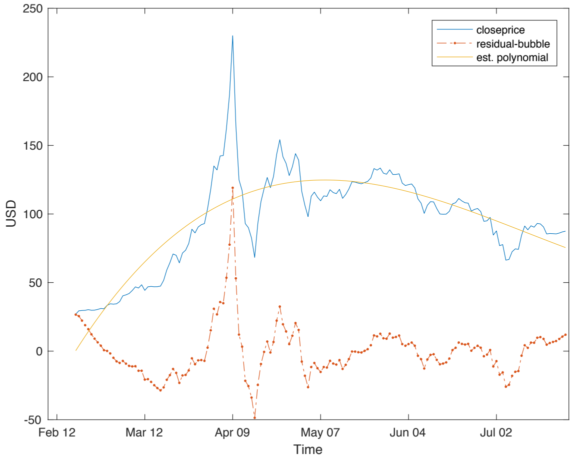


Figure 3. Bitcoin/USD price decomposition

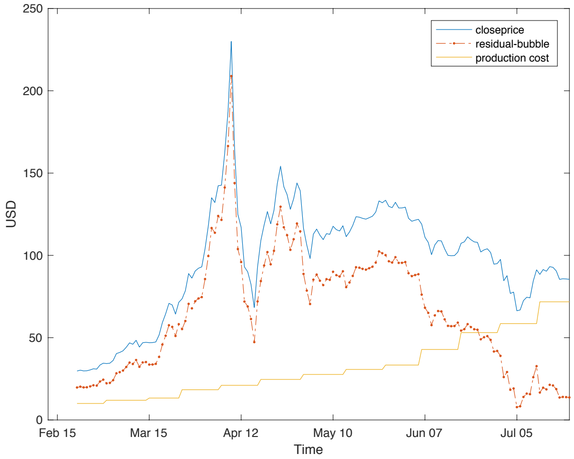


Figure 4. Bitcoin/USD price vs. production cost

5 Estimated models

The following part of the study undertakes a mixed causal/noncausal analysis by estimating MAR models on the BTC/USD price and on the bubble component according to the two different definitions of the fundamental part. As already discussed in the introduction it is expected that the *forward looking* dependence is stronger in the isolated bubble component than in the observed price. The model specifications in what follows are chosen by applying *information criteria* which are useful tools to select the number of lags (and leads) to be included in the model. The information criteria hereby considered are the *Akaike information criterion* (AIC), the *Bayesian information criterion* (BIC) and the *Hannan-Quinn information criterion* (HQ) (for a general review see the book by Hamilton [12]). Once the number of lags/leads have been detected, models are estimated by maximizing the approximated log-likelihood function based on the Student's *t* density function for the error term; a detailed description of the procedure may be found in Hecq et al. [14]. The related Matlab routines used in this work are kindly provided by the authors of the above quoted paper.

Table 1. AR(1) model's estimated parameters

AR(1) Model		***t* distribution**	
φ_1	Std. Dev.	λ	ν
0.8066	0.0234	4.3928	2.5013

Table 2. BDS test results, purely noncausal model AR(1).

m	w	$p - value$	m	w	$p - value$
2	5,978547545	1,13E-09	9	13,3666165	0
3	6,525463574	3,39E-11	10	14,65204326	0
4	7,420797806	5,82E-14	11	15,91260972	0
5	8,615265114	0	12	17,42915916	0
6	9,743131337	0	13	19,3469674	0
7	10,83386529	0	14	21,49415105	0
8	12,07560832	0	15	24,05813227	0

5.1 Noncausal analysis of the bubble component

Firstly is considered a *strictly noncausal* AR(1) (forward looking):

$$y_t = \varphi_1 \, y_{t+1} + \epsilon_t.$$

where ϵ_t are *iid* Student's *t* distributed errors, with location 0 and scale parameter λ, $\epsilon_t \sim (0, \lambda)$. Estimated parameters are reported in Table 1. The residuals of the models are shown in Fig 5. In order to test the model's goodness of fit, the results of the BDS test (Brock, William, Davis Dechert & Scheinkman, 1987) [16], used to test whether the residual are truly a sequence of *iid* Student's *t* random variables, are reported in Table 2. The test fails to accept the null hypothesis of *iid* distributed residuals, this implies that the present model must be discarded.

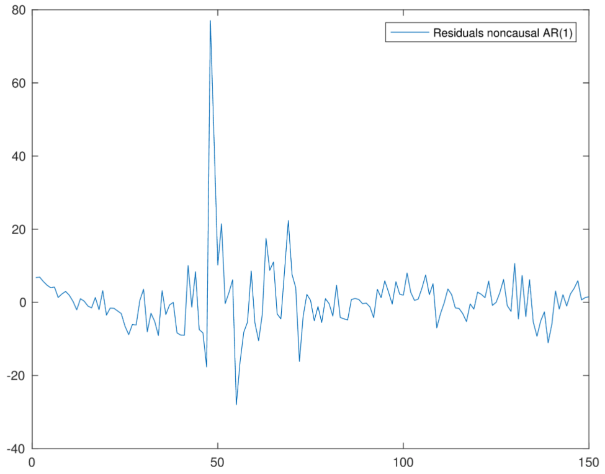


Figure 5. Noncausal AR(1) model residuals

Table 3. Information Criteria

p	BIC	AIC	HQ	p	BIC	AIC	HQ
0	6,0119	5,9565	5,9649	5	4,8862	4,5537	4,6042
1	4,7057	4,5949	4,6117	6	4,914	4,526	4,585
2	4,6823	4,5161	4,5413	7	4,9494	4,506	4,5734
3	4,7513	4,5296	4,5633	8	4,987	4,4882	4,5639
4	4,818	4,5409	4,583				

Table 4. MAR(1,1) estimated parameters.

Parameter	Estimate	Confidence bounds	
ϕ_1	0.5255	0.4585	0.5925
φ_1	0.6503	0.5897	0.7110

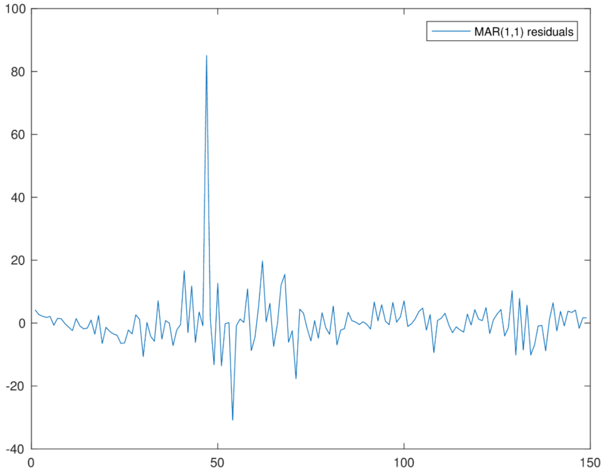


Figure 6. *Mixed causal-noncausal* MAR(1,1) residuals

Table 5. BDS test results for the MAR(1,1) model residuals on y_t.

m	w	$p - value$	m	w	$p - value$
2	1,703389269	0,0442	9	6,924875969	2,18E-12
3	2,249277819	0,0122	10	7,294838916	1,50E-13
4	3,342684301	4,15E-04	11	7,639695322	1,09E-14
5	4,384330303	5,82E-06	12	8,143710723	2,22E-16
6	5,191782337	1,04E-07	13	8,892820344	0
7	5,887119414	1,96E-09	14	9,500629834	0
8	6,412692984	7,15E-11	15	10,18321732	0

5.1.1 Mixed causal-noncausal AR model

The following specification of the model is derived by the suggestions of the *information criteria*, these are very useful tools to determine the time dependencies to be included in the model, *i.e.* they are used to determine the order of the autoregressive polynomial (see equation 1) p. The information criteria hereby considered are the *Akaike information criteria* AIC, the *Bayesian information criteria* and the *Hannan-Quinn* information criterion HQ [14], Hecq et al. [14] show that simulation results would favour the use of BIC. As reported in Table 3 the information criteria suggest setting $p = 2$.

When $p = 2$ the estimated *Mixed causal-noncausal* model is a MAR(1,1):

$$(1 - \phi_1 L)\,(1 - \varphi_1 L^{-1})\,y_t = \varepsilon_t.$$

Table 4 shows the estimated parameters of the model. Fig. 6 shows the sequence of the MAR(1,1) model's estimated residuals $\hat{\varepsilon}_t$.

As shown in Table 5, the BDS test for independence fails to accept the null hypothesis of *iid* distributed residuals for most of the tested *embedded dimensions*, thus suggesting to discard the model just now estimated.

Table 6. Information Criteria, MAR model on $rate_t$

p	BIC	AIC	HQ	p	BIC	AIC	HQ
0	7,0358	6,9803	6,9888	5	4,9587	4,6262	4,6767
1	4,7483	4,6374	4,6543	6	5,0137	4,6257	4,6847
2	4,7546	4,5883	4,6136	7	5,0781	4,6348	4,7021
3	4,8215	4,5998	4,6335	8	5,0705	4,5716	4,6474
4	4,8906	4,6135	4,6556				

5.2 Noncausal analysis of the observed price BTC/USD

Since the interpretation of the aforementioned estimated parameters can be rather misleading and therefore hard to be extended to the market reality, given the arbitrary choice for

Table 7. MAR(0,1) estimated autoregressive parameters on $rate_t$

Parameter	Estimate	Confidence bounds	
ϕ_1	0.9809	0.9740	0.9878

Table 8. Estimated parameters of the Student's t error distribution, MAR(0,1) model on $rate_t$

λ	ν
3.3073	1.4863

Table 9. BDS test results for the MAR(0,1) model residuals

m	w	H_0	m	w	H_0
2	6,45069	1	9	16,22165	1
3	8,26635	1	10	18,05194	1
4	9,30074	1	11	20,14559	1
5	10,38450	1	12	22,50443	1
6	11,66036	1	13	25,59976	1
7	13,03694	1	14	29,31705	1
8	14,59404	1	15	33,57504	1

Table 10. MAR(1,1) estimated autoregressive parameters on $rate_t$

Parameter	Estimate	Confidence bounds	
ϕ_1	0.9747	0.9650	0.9844
φ_1	0.2781	0.2077	0.3485

the fundamental component, it is of great interest to estimate the MAR model directly on the BTC/USD time series ($rate_t$).

The aforementioned information criteria, in application to the BTC/USD time series, suggest to set the order of the autoregressive polynomial to $p = 1$ (see Eq. 1) or $p = 2$ depending on the selected criterion (see Table 6).

5.2.1 Estimated MAR model, case $p = 1$

When $p = 1$ the estimated model that best fits the observed time series $rate_t$ is a *purely causal* AR(1,0). Table 7 and Table 8 display the estimated parameters of the autoregressive polynomial and of the error distribution, respectively.

Since the distribution's degrees of freedom $\nu = 1.4863 < 2$, then the estimated sequence of error terms $\hat{\epsilon}_t$ cannot be likened to the case $\epsilon_t \sim iid\ t_\nu\,(0, \lambda)$, given the fact that when $\nu < 2$ the expected value of the distribution is not defined. Anyhow the BDS test (Table 12) for independence does not accept the null hypothesis of *iid* distributed residuals, thus suggesting once again to discard the estimated model.

Table 11. Estimated parameters of the Student's t error distribution, MAR(0,1) model on $rate_t$

λ	ν
3.3901	1.6043

Table 12. BDS test results for the MAR(1,0) model residuals on $rate_t$.

m	w	$p - value$	m	w	$p - value$
2	6,450686379	5,57E-11	9	16,22165279	0
3	8,266350334	1,11E-16	10	18,05194201	0
4	9,300742867	0	11	20,14558908	0
5	10,38450476	0	12	22,50443038	0
6	11,66035838	0	13	25,5997597	0
7	13,03694212	0	14	29,31704918	0
8	14,59403959	0	15	33,57504129	0

Table 13. BDS test results for the MAR(1,1) model residuals on $rate_t$.

m	w	$p - value$	m	w	$p - value$
2	5,121150647	1,52E-07	9	15,28926444	0
3	7,537452115	2,40E-14	10	16,97636905	0
4	8,912767967	0	11	18,98947641	0
5	10,07953357	0	12	21,39229563	0
6	11,15351875	0	13	24,08437913	0
7	12,36324983	0	14	27,23839452	0
8	13,78408522	0	15	30,78448895	0

5.2.2 MAR model, p=2

When $p = 2$ the estimated model is a *Mixed causal-noncausal* MAR; estimates of the model are displayed in Table 10 and 11. Once again the estimated t distribution's degrees of freedom is $\nu = 1.6043 < 2$, therefore the estimated sequence of error terms cannot be likened to the case $\epsilon_t \sim iid\ t_\nu\,(0, \lambda)$, suggesting to discard the model once again. In any case, the BDS test(Table 12) for independence does not accept the null hypothesis of *iid* distributed residuals, thus suggesting once again to discard the estimated model.

5.3 Residual analysis

To sum up, MAR models are estimated directly on the BTC/USD time series ($rate_t$) and then on the bubble part (y_t, yc_t); the aforementioned information criteria suggest to set the order of the autoregressive polynomial to $p = 1$ or $p = 2$ depending on the selected criterion, both for the BTC/USD rate and for the bubble terms. In the former case, $p = 1$, a *strictly causal backward looking* AR(1) is the preliminary reference specification for both the full rate $rate_t$ and the bubble component yc_t whereas a *strictly non-causal forward looking* AR(1) is the preliminary reference for the bubble component y_t. For the latter case, $p = 2$, a MAR(1,1) model is found to be fitting all the time series $rate_t$, y_t and yc_t.

The estimation results are reported in Table 14.

Table 14. MAR(r,s) estimated parameters on $rate_t$, y_t & yc_t

T. Series	MAR(r,s)	Par.	Est.	Conf. bounds		Par.	Est.	Conf. bounds	
$rate_t$	MAR(1,0)	ϕ_1	0.9809	0.9740	0.9878	φ_1	-	-	-
	MAR(1,1)	ϕ_1	0.9747	0.9650	0.9844	φ_1	0.2781	0.2077	0.3485
y_t	MAR(0,1)	-	-	-	-	φ_1	0.8066	0.8028	0.8103
	MAR(1,1)	ϕ_1	0.5255	0.4585	0.5925	φ_1	0.6503	0.5897	0.7110
yc_t	MAR(1,0)	ϕ_1	0.9803	0.9702	0.9904	φ_1	-	-	-
	MAR(1,1)	ϕ_1	0.3424	0.2604	0.4245	φ_1	0.9396	0.9216	0.9576

It is evident from the results in Table 14 that there is a very strong *backward looking* dependence in one lagged value, for the BTC/USD rate and for the bubble component yc_t; conversely, for the isolated bubble term y_t, there is a very strong *forward looking* dependence in one led value.

The estimation of a *Mixed causal/non-causal* MAR(1,1) gives further insights on the *backward* and *forward* dependence; outcomes are summed up in Table 14 respectively for the full rate $rate_t$, the bubble term y_t and the bubble term yc_t.

Particularly interesting is the difference in the parameter ϕ_1 and φ_1 when estimating the MAR(1,1) model separately on the bubble component y_t and on the original time series $rate_t$. As shown in Table 14 the non-causal parameters (*forward looking*) φ are stronger in the bubble component y_t than in the observed price $rate_t$, whereas the causal parameter ϕ is much stronger in the observed price $rate_t$ than in the bubble component y_t. This is consistent with the conjecture made in the introduction, that the speculative bubble is rather a forward looking phenomena than a past one, since the *forward looking* estimated parameters on the bubble part are stronger than the ones on the observed BTC/USD price $rate_t$. This evidence is strengthen by the MAR(1,1) estimated parameters on the bubble part yc_t, indeed it can be noticed that the value of the forward looking and backward looking components almost trade places when estimating the model on the full price time series $rate_t$ and the bubble component yc_t respectively. As mentioned in Section 3.2, if the model is correctly specified then the model residuals ϵ_t should be a sequence of *Independent Identically Distributed* Student's t observations. In this study the *IID* hypothesis is tested through the BDS test for independence. This test is based on the correlation dimension, with m *embedded dimension*, since it can be shown [16] that the test statistic w is asymptotically normally distributed $\sim \mathcal{N}(0, 1)$, it is quite feasible to obtain $p - values$. The Tables 15 and 16 reporting the outcome of performing such test on the residuals ϵ_t of the yc_t estimated models. As shown in Table 5 it can be noticed that the only model for which the null hypothesis of *IID* residuals cannot be rejected is the MAR(1,1) model on y_t, and only for $m = 1$ or $m = 2$, depending on the selected confidence bound width.

Table 15. BDS test results for the MAR(1,0) model residuals on yc_t.

m	w	$p - value$	m	w	$p - value$
2	5,528256527	1,62E-08	9	12,29309902	0
3	7,009012288	1,20E-12	10	13,36389728	0
4	7,773551059	3,77E-15	11	15,15913039	0
5	8,506760111	0	12	17,1200305	0
6	9,287013257	0	13	19,26464437	0
7	10,17111148	0	14	21,78528929	0
8	11,24555355	0	15	24,56128819	0

Table 16. BDS test results for the MAR(1,1) model residuals on yc_t.

m	w	$p - value$	m	w	$p - value$
2	4,236007075	1,14E-05	9	10,32346963	0
3	5,315716933	5,31E-08	10	11,13612446	0
4	6,272423899	1,78E-10	11	12,51271091	0
5	7,320422719	1,24E-13	12	14,01078341	0
6	8,132958153	2,22E-16	13	15,68841368	0
7	8,838578863	0	14	17,59084596	0
8	9,612705164	0	15	19,49717776	0

6 Conclusions

This study undertook a *Mixed causal-noncausal* analysis of the BTC/USD exchange rates time series, over the period February-July 2013, to test whether the bubble effect disentangled on observed data may be explained by a *forward looking* behaviour of the economic agents. In the introduction it was noticed that given the system's monetary issuance, the exchange rate of one Bitcoin with respect to a traditional currency should be influenced by agents's future expectations and that classical ARIMA models, *backward looking* by definition, are not suitable to describe the dynamics of the Bitcoin price given the fact that the only time dependence admitted by these model regards the past. *Mixed backward forward looking* MAR models are hence considered both for the BTC/USD exchange rate and for the isolated bubbles.

The conjecture underlying this study is that the forward looking parameters should be stronger in the bubble part than in the observed price. Indeed this turns out to be the case, when estimating the model on the observed data, however the residuals analysis, conducted by performing the BDS test for independence, suggests not to consider this models valid but for one case (partially). Since the results of this test are asymptotical (for $n \to \infty$) and given the low entity of the residuals a more extensive residual analysis could be performed in order to assess the capability of the chosen model to describe the dynamics of BTC/USD rate and/or the isolated bubble term (y_t, yc_t). Several techniques are available such as the classical *Ljung-Box-Q* test on residuals autocorrelation (see [12]). Although the focus of this study is not to come across the true Data Generating Process for the Bitcoin, a deeper investigation of this issue is beyond the scope of the present study and will be tackled in future research.

In the future we plan to evaluate the possibility of proposing cross-evaluation techniques, and propose complementary validation with regression metrics such as RMSE, MAE, RMSD and others.

Acknowledgments

This work is supported by project "Agrichain" (funded by Fondazione Cassa di Risparmio di Perugia 2018-2020) and project "ASIA" (funded by INDAM-GNCS).

References

[1] Stefano Bistarelli, Alessandra Cretarola, Gianna Figà-Talamanca, Ivan Mercanti, and Marco Patacca. Is arbitrage possible in the bitcoin market? (work-in-progress paper). In *GECON*, volume 11113 of *Lecture Notes in Computer Science*, pages 243–251. Springer, 2018.

[2] Stefano Bistarelli, Ivan Mercanti, and Francesco Santini. A suite of tools for the forensic analysis of bitcoin transactions: Preliminary report. In *Euro-Par Workshops*, volume 11339 of *Lecture Notes in Computer Science*, pages 329–341. Springer, 2018.

[3] Stefano Bistarelli, Matteo Parroccini, and Francesco Santini. Visualizing bitcoin flows of ransomware: Wannacry one week later. In *ITASEC*, volume 2058 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018.

[4] Stefano Bistarelli and Francesco Santini. Go with the -bitcoin- flow, with visual analytics. In *ARES*, pages 38:1–38:6. ACM, 2017.

[5] Olivier Blanchard. Speculative bubbles, crashes and rational expectations. *Economics Letters*, 3(4):387–389, 1979.

[6] George Edward Pelham Box and Gwilym Jenkins. *Time Series Analysis, Forecasting and Control.* Holden-Day, Inc., San Francisco, CA, USA, 1990.

[7] F.Jay Breid, Richard A Davis, Keh-Shin Lh, and Murray Rosenblatt. Maximum likelihood estimation for noncausal autoregressive processes. *Journal of Multivariate Analysis*, 36(2):175 – 198, 1991.

[8] Gerald Dwyer. The economics of bitcoin and similar private digital currencies. *Journal of Financial Stability*, 17(C):81–91, 2015.

[9] David García, Claudio Juan Tessone, Pavlin Mavrodiev, and Nicolas Perony. The digital traces of bubbles: feedback cycles between socioeconomic signals in the bitcoin economy. *CoRR*, abs/1408.1494, 2014.

[10] Christian Gourieroux and Joann Jasiak. Filtering, prediction and simulation methods for noncausal processes. *Journal of Time Series Analysis*, 37(3):405–430, 2016.

[11] Christian GouriÃĪroux and Jean-Michel Zakoian. Explosive Bubble Modelling by Noncausal Process. Working Papers 2013-04, Center for Research in Economics and Statistics, February 2013.

[12] James D Hamilton. Time series econometrics. *Princeton U. Press, Princeton*, 1994.

[13] Adam Hayes. Cryptocurrency value formation: An empirical study leading to a cost of production model for valuing bitcoin. *Telematics and Informatics*, 34:1308–1321, 11 2017.

[14] Alain Hecq, Lenard Lieb, and Sean Telg. Identification of mixed causal-noncausal models in finite samples. *Annals of Economics and Statistics*, 123/124:307–331, 2016.

[15] Andrew Hencic and Christian Gouriéroux. Noncausal autoregressive model in application to bitcoin/usd exchange rates. In *Econometrics of Risk*, volume 583 of *Studies in Computational Intelligence*, pages 17–40. Springer, 2015.

[16] Ludwig Kanzler. Very fast and correctly sized estimation of the bds statistic. *Available at SSRN 151669*, 1999.

[17] Ladislav Kristoufek. What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis. *PLOS ONE*, 10(4):1–15, 04 2015.

[18] Markku Lanne and Pentti Saikkonen. Modeling expectations with noncausal autoregressions. *Available at SSRN 1210122*, 2008.

[19] Markku Lanne and Pentti Saikkonen. Noncausal autoregressions for economic time series. *Journal of Time Series Econometrics*, 3(3), 2011.

Unsupervised Context Extraction via Region Embedding for Context-Aware Recommendations

Padipat Sitkrongwong
The Graduate University for Advanced Studies
(SOKENDAI)
Tokyo, Japan
padipat@nii.ac.jp

Atsuhiro Takasu
National Institute of Informatics / SOKENDAI
Tokyo, Japan
takasu@nii.ac.jp

ABSTRACT

Many context-aware recommendation methods extract contexts from reviews using supervised methods. However, this requires the optimal values for contexts to be predefined, which is not a trivial task. Although some approaches have avoided this by utilizing unsupervised methods, the extracted contexts have been limited to a unigram format. Moreover, most methods consider only the influence of context on the entire dataset, ignoring the fact that context might be relevant to individual users or items unequally. This work proposes a novel unsupervised context extraction method that uses predictive models for future ratings. Unlike previous work, we extract context from reviews automatically in the form of skip-grams by applying a region embedding technique. The predictive models utilize the interaction between contexts and users (and items) to model their influence on ratings. Experiments demonstrate that our models can outperform existing review-based recommendations that ignore contexts.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Computing methodologies → Information extraction**; *Unsupervised learning*; *Neural networks*.

KEYWORDS

context-aware recommender systems, region embedding, review mining, opinion mining

ACM Reference Format:
Padipat Sitkrongwong and Atsuhiro Takasu. 2019. Unsupervised Context Extraction via Region Embedding for Context-Aware Recommendations. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331076.3331120

1 INTRODUCTION

Recommender systems were devised to provide personalized recommendations about specific items to individual users. The most common approach to making recommendations is to exploit the users' past preferences about items, such as their ratings, to create a predictive model for their future rating of unseen items. In addition to using rating data, context-aware recommenders offer more effective recommendations by taking into account contextual information (or simply "context"). Context such as location, time or weather can have a major influence on users' decisions when they are choosing items. For example, if a user is seeking a hotel for a summer vacation, the recommendation engine should suggest hotels in a beach area, rather than in mountainside ski resorts. Incorporating such contexts has been shown to provide more accurate predictions than standard context-free recommender systems [2].

Although context can improve recommendations, obtaining it is not trivial. In traditional recommendation schemes, users review items they have previously chosen and assign rating scores to indicate their preference levels for those items. Context is rarely provided. To obtain such information, early approaches to context-aware recommenders collected context by explicitly asking users to supply it. Specifically, in addition to their ratings on items, users were asked to select a context from a predefined list of context options. Collecting context data in this way is not only expensive but also not useful in real-world scenarios where most users have no intention of providing such information. Therefore, many context-aware methods try to infer the context from additional sources of data [9, 13, 23]. In recent years, the most popular source of context has been user-generated reviews [7, 15, 17].

In reviews, users can express opinions about their experience and level of satisfaction with the items concerned, which can, therefore, be a rich source of context data [7]. However, the context in reviews has to be recognized as such before it can be used. Two common approaches to this task are to using text-mining techniques or to define contexts as latent variables. The text-mining approach [6, 10, 12] utilizes techniques such as string matching to identify and extract words in a review that match a predefined list of context strings. Determining the optimal values of contexts for a specific domain then becomes the main challenge in this approach because it can significantly affect the quality of the recommendations. To address this issue, some approaches have applied unsupervised techniques to extract context from reviews [5, 19]. However, the data extracted by this work has been restricted to a unigram format (e.g., "night" or "friend"). More realistically, the context might well involve more than just one word, in the form of n-grams such as "business trip" or "king-sized bed."

After obtaining the context, it is then important to separate the relevant context from the irrelevant context. Because not all context will be related to the objective of the recommendation, including irrelevant information could degrade the recommendation quality

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331120

[2]. Several methods define relevant context items as those that have a significant influence on ratings' distributions [2, 14, 18]. However, this is applicable only to predefined types of contexts that have static values and are of fixed size. Moreover, most methods identify relevant context based on its influence on an entire dataset. In reality, the relevance of an element of context might well depend on individual users' preferences and the specific target items' features, which would, therefore, influence the ratings in particular reviews.

In this work, we propose a novel method for defining, extracting and representing context from review data, together with an efficient method to utilize context for personalized rating predictions. We define context as any word in a region of text that has an influence on the distribution of ratings. Such words can be in unigram, n-gram or even skip-gram format, provided that they reside within the same region of text. By applying region embedding with the local context unit proposed by [21], the positions of context words in a text region can be emphasized as those that contribute the highest variance in ratings. As a result, contexts can be represented by the region embeddings that capture their influence on the rating distributions for an entire review dataset. To generate a rating prediction, we model the influences of context items on a particular review's rating based on their past interactions with an individual user and/or item. Based on these interactions, we propose four predictive models for rating prediction. Our experiments demonstrate that our proposed method is more accurate than the review-based recommendation methods that utilize only word embeddings and do not consider the context in making rating predictions.

2 RELATED WORK

Our proposed method relates mainly to two subcategories of recommender systems, namely context-aware and review-based recommenders.

2.1 Context-Aware Recommenders

In context-aware recommender systems, context is usually defined as "any information that can be used to characterize the situation of an entity" [1]. Based on this definition, [8] classified context into *representational* and *interactional* approaches. Most of the proposed context-aware methods adopt the representational approach, whereby contexts are defined by predefined sets of values, and their structures are static (e.g., the context *Time* might be defined with the values {"weekday," "weekend"}). Defining optimal values for the predefined contexts, however, is not a trivial task, and depends on the specific recommendation domains.

The interactional approach, on the other hand, assumes that user behaviors are influenced by context, but the contexts themselves are not necessarily observable. In this approach, therefore, unsupervised techniques are required to extract and represent contexts. In one such example, [9] applied topic modeling to represent context as sequences of latent topics that capture changes in users' interests. In another example, [23] infers context from mobile-sensor data and represents it in a latent low-dimensional space using unsupervised deep learning techniques. Finally, [13] treats social media's new feeds as the context for recommending the top-k relevant advertisements to users. This interactional approach has the advantage of not having to predefine context values, which enables the discovery of hidden or unobserved contexts, and is therefore applicable to a wider range of recommendation domains than the representational approach.

To produce effective recommendations, only relevant context should be taken into account. For example, the context *Companion* (e.g., "friend" or "family") would probably be more relevant to users' decisions on choosing movies to watch, than the context *Season* (e.g., "summer" or "winter"). To identify the relevant contexts, [3] conducted a user survey that let users imagine rating items in different contexts and evaluated the influence in each case. In addition to such manual methods, [2, 14, 18] applied statistical testing such as the paired t test or Pearson's chi-squared test to detect the relevant context items as those that contributed significant differences to the distribution of ratings for each context item. For example, the context *Companion* would be considered as relevant if most users who watched movies with friends gave higher overall ratings while those who watched with family gave lower ratings. Alternatively, [18] classified the influence of context into two levels, namely the *population* and the *individual* level. The relevant context at the population level influences the rating distributions for the entire dataset, whereas the relevant context at the individual level influences the ratings for individual users or items. Most methods effectively identify relevant context at the population level, using techniques such as [2, 14]. However, some methods identify relevant context at the individual level. For example, [4] predicted ratings by modifying a matrix factorization to model the relationship between contexts and items. In our previous work [22], we proposed a latent probabilistic model that captured the interactions between relevant context and the user (and item) classes.

Most methods for identifying the relevant contexts are only applicable to a representational approach (i.e., predefined, static and fixed-size contexts). In this work, we propose using an interactional approach to extract the relevant context from review data, based on the idea of two influence levels for context proposed in [18]. In this paper, we will refer to the influence at the population level as the *global* influence and to the influence at an individual level as the *local* influence.

2.2 Review-based Recommenders

In recent years, many recommendation techniques have tried to incorporate user-generated reviews as the main resource for modeling user preferences [7]. To produce recommendations using the contents of reviews, some methods have exploited word-embedding techniques [15, 17]. These methods assume that the low-dimensional representations of words such as those in Word2Vec [16] or GloVe [20] can be used to build and represent the user and target-item profiles. In [17], these profiles were computed by summing the word embeddings of all words in all reviews associated with the user and item. Inspired by this idea, [15] combined the ratings as the weights for each review to create the profiles. Specifically, the preference profile for user u_i: $\mathrm{S}_i \in \mathbf{R}^{1\times h}$ and the characteristic profile for item v_j: $\mathrm{S}_j \in \mathbf{R}^{1\times h}$, where h is the embedding size, can be computed respectively by Equations 1 and 2.

$$\mathrm{S}_i = \sum_{q_{(i,j)} \in Review_{u_i}} \sum_{w_t \in q_{(i,j)}} w2v(w_t) \times pref_{(i,j)}, \tag{1}$$

$$\mathrm{S}_j = \sum_{q_{(i,j)} \in Review_{v_j}} \sum_{w_t \in q_{i,j}} w2v(w_t) \times pref_{(i,j)}. \tag{2}$$

Here, the parameter $pref_{(i,j)}$ is the rating by user u_i of item v_j, $Review_{u_i}$ and $Review_{v_j}$ are the sets of u_i's and v_j's reviews, respectively. The function $w2v(w_t)$ maps the word w_t to its embedding representation, which can be learned prior by any word-embedding technique. For rating prediction, [15] further combines the user and target-item profiles with the latent factors from matrix factorization as expressed in Equation 3.

$$pref_{(i,j)} \approx \alpha \mathrm{U}_i^T \mathrm{V}_j + (1-\alpha)(\mathrm{U}_{s,i}^T \mathrm{S}_i + \mathrm{V}_{s,j}^T \mathrm{S}_j) \tag{3}$$

Let $|latent|$ denote the number of latent dimensions. The model then applies stochastic gradient descent to learn jointly, for user u_i and item v_j, the latent factors $\mathrm{U}_i, \mathrm{V}_j \in \mathbf{R}^{1\times|latent|}$ and the semantic latent factors $\mathrm{U}_{s,i}, \mathrm{V}_{s,j} \in \mathbf{R}^{1\times h}$. The rating $pref_{(u,i)}$ is then computed as a linear combination of the latent factors and the semantic latent factors, with α being the parameter controlling the trade-off.

Note that all previously mentioned methods exploit pretrained word embeddings of all words in reviews to represent the user and item profiles. We believe this is inefficient because some words have a more significant effect on user preferences than others. For example, stopwords such as "is," "a" or "the" are likely to affect ratings less than opinion words (e.g., "great" or "not") or context words (e.g., "service" or "friendly"). Moreover, although the word embeddings are pretrained to capture the semantic meaning of words, their relationship with the ratings is not captured.

Many context-aware methods have been proposed that exploit review-based recommendations to make use of the rich and valuable contextual information contained in reviews. For example, [6, 10, 12] applied supervised text-mining techniques to extract contexts from reviews. However, these methods adopt the representational approach, which requires contexts to be predefined. Some methods have tried to extract review contexts using the interactional approach [5, 19]. They partitioned the reviews into contextual reviews (containing contexts) and non-contextual reviews. The contexts were then extracted as those words or topics that occurred most often in the contextual reviews. Although their method has the advantage of not having to predefine the values of contexts, their extracted contexts are based on the "bag-of-words" approach, where contexts are in the form of unigrams. Because reviews are written as free-form text, we should take this opportunity to explore and utilize the contexts that could possibly be constructed from more than just one word.

In this work, our contexts are represented by skip-grams of words that occupy the same text region. Their embedding representations are learned directly to capture their relationship with the ratings. Consequently, those words with more influence on the rating distributions will have more impact on the rating predictions.

3 PROPOSED METHOD

In this section, we provide a detailed explanation of our proposed method. The workflow of our model is illustrated in Figure 1. The method comprises two main parts, namely context extraction and rating prediction. In the context extraction part, we first identify the candidates for context words and extract all text regions containing them, based on their global influence on the distributions of review ratings. The text regions and their rating distributions are then fed into a neural network to learn the word embeddings and local context units, which are then used to compute the region embeddings. In the rating prediction part, we model the local influences of contexts on a review rating based on the past interactions between their associated region embeddings with an individual user and/or item. Based on these interactions, we propose four predictive models for rating prediction, each of which is best suited to a particular setting for the review data.

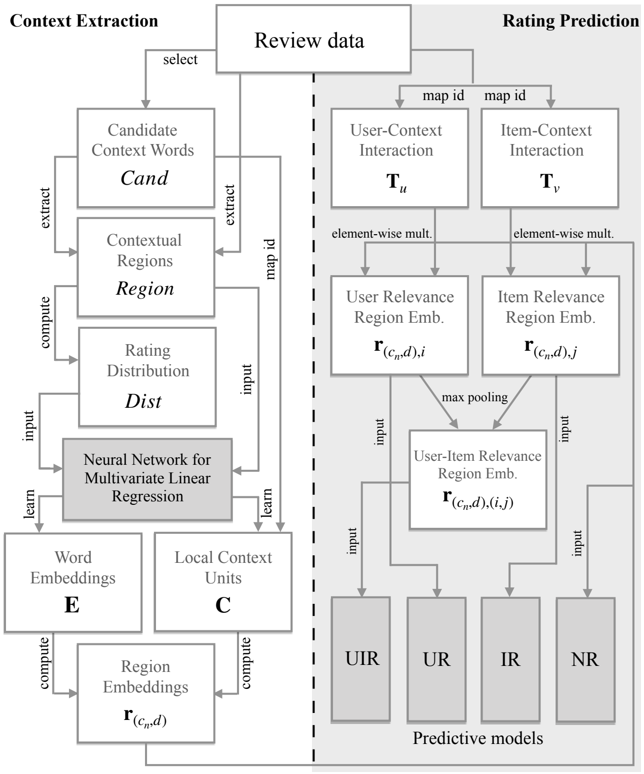
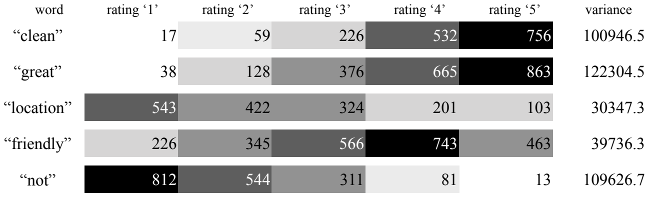


Figure 1: A workflow architecture of the proposed method.

3.1 Context Extraction

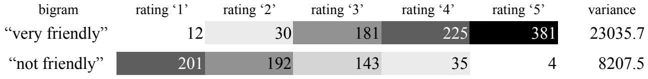
We first present our context extraction method, which comprises three main steps, namely identifying the candidate context words, extracting the contextual regions and learning the region embeddings.

3.1.1 Identifying the Candidate Context Words. We derive the definition from [18], where relevant contexts are defined as those that contribute to explaining the variance in ratings. By applying this definition to review data, a context can be any word in the reviews that influences the distribution of ratings (i.e., its variance). For example, suppose there are 100 reviews containing the word "friendly" that use a 1–5 rating scale, with 60 having a rating of "5," 30 having a rating of "4" and the remaining 10 having ratings of "1," "2" and "3." This means that "friendly" implicitly influences the



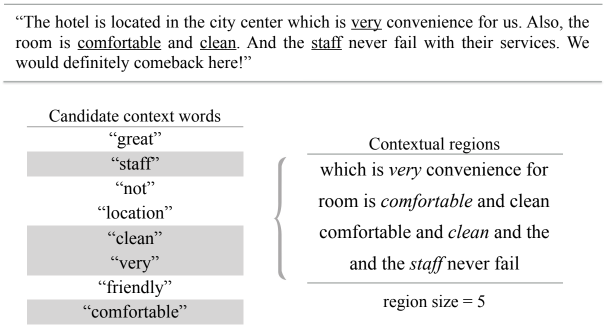
word	rating '1'	rating '2'	rating '3'	rating '4'	rating '5'	variance
"clean"	17	59	226	532	756	100946.5
"great"	38	128	376	665	863	122304.5
"location"	543	422	324	201	103	30347.3
"friendly"	226	345	566	743	463	39736.3
"not"	812	544	311	81	13	109626.7

Figure 2: Example of word-rating co-occurrences and their corresponding variances.



bigram	rating '1'	rating '2'	rating '3'	rating '4'	rating '5'	variance
"very friendly"	12	30	181	225	381	23035.7
"not friendly"	201	192	143	35	4	8207.5

Figure 3: Example of rating co-occurrences with two bigrams containing the word "friendly"



"The hotel is located in the city center which is very convenience for us. Also, the room is comfortable and clean. And the staff never fail with their services. We would definitely comeback here!"

Figure 4: Extracting the contextual regions from a review

distribution of the reviews' ratings toward high ratings, and it can, therefore, be considered a candidate for the set of relevant contexts. Therefore, our first step in extracting the contextual information from reviews is to identify the set of words that have influences on the distributions of ratings over all reviews. We call this set of words the *candidate context words*.

To extract the candidate context words, we first create a word-rating co-occurrence matrix by counting the number of times each word in reviews occurs with each rating. Next, we calculate the variance from the frequency distribution of ratings for each word. Finally, only those words having a calculated variance above the minimum variance threshold min_{var} are selected as candidate context words and are stored in the candidate list *Cand*. An example of the word-rating co-occurrence matrix and the corresponding variances is shown in Figure 2.

In Figure 2, each cell contains a word frequency under each rating value. For instance, "clean" occurred in reviews with rating "3" 226 times. The distributions of ratings can be visualized by the cell's grayscale shading, which represents the densities of the word frequencies from the highest (black) to the lowest (white). From this figure, we can observe, for example, that the word "clean" is distributed toward high ratings, while "location" is distributed toward low ratings. Note that, in addition to context words (e.g., "clean"), opinion or sentiment words such as "great" or "not" also have a great influence on the distributions of ratings. Therefore, we also include such words as candidate context words.

3.1.2 Extracting Contextual Regions and Their Rating Distributions.

Depending solely on the candidate context words might not be sufficient to cover the variety of influences of contexts on the distributions of reviews' ratings. This is because neighboring words that are often written together with those candidate context words might significantly alter the ways they influence the distributions of ratings. For example, Figure 3 shows the example of the co-occurrence of two bigrams "very friendly" and "not friendly." Although they are generated from the same candidate context word "friendly," their rating distributions are totally different (one is very positive and the other is very negative). This is because those nearby words might be opinion, sentiment or other words that could change the semantic meanings of the candidate context words, and therefore influence their rating distributions. For this reason, neighboring words are crucial and should be incorporated with the candidate context words to model the influences of contexts effectively.

Consider a candidate context word $c_n \in Cand$. We define the nearby words of c_n as any word w_t that occupies the same text region of c_n, i.e., $w_t \in region_{(c_n,d)}$, where d is the window size for a region of length $2 \times d + 1$. Note that w_t can be in any position in $region_{(c_n,d)}$, and it does not necessarily have to be directly adjacent to c_n. This takes account of the different writing styles users may adopt for the same meaning in writing reviews. For example, "the *staff* are *friendly*" and "they have *friendly staff*" both indicate the same context "friendly staff."

We, therefore, define a context as any word in a region of text that has an influence on the distributions of ratings. Such words can be in unigram, n-gram or even skip-gram form, provided they occupy the same region. To identify the positions of these words, we first need to extract their associated regions, namely *contextual regions*. For each c_n, we extract all contextual regions of size $2\times d+1$ from all reviews containing c_n. Figure 4 shows an example of the contextual regions of size 5 extracted from one review. Given that this review contains four candidate context words, four contextual regions are extracted.

After all contextual regions are extracted, we store them in the list of contextual regions *Region*. For each contextual region with the index m: $region^{(m)}_{(c_n,d)} \in Region$, we need to identify the positions of the words that, along with c_n, contribute to the highest variance in the rating distributions. This can be achieved in the same way as identifying the candidate context words. Specifically, for $region^{(m)}_{(c_n,d)}$, we first generate all skip-grams of size ρ (where $\rho \leq 2 \times d + 1$) that include c_n. We then count the number of times each skip-gram occurs in the same region with each rating and calculate the variance from the frequency distribution of ratings.

Figure 5 illustrates the process of identifying the highest contributed variance skip-gram of size $\rho = 2$ for the region "room is *comfortable* and clean." If the skip-gram of c_n and w_t yields the highest variance and is more than min_{var}, w_t will be chosen as a part of the context for the region, along with c_n. Finally, the rating distribution of the skip-gram that yields the highest variance is selected as the label $dist^{(m)}_{(c_n,d)}$ to represent the rating distribution of that region. In Figure 5, the rating distribution of the skip-gram ("comfortable,"

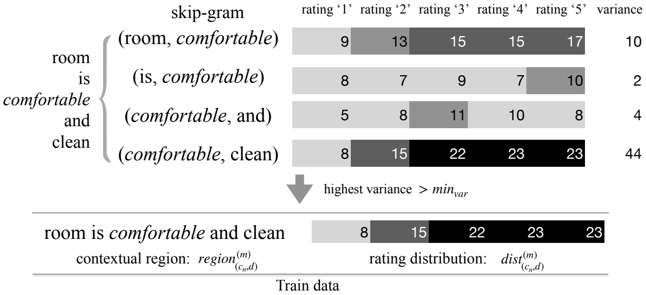

skip-gram
rating '1'
rating '2'
rating '3'
rating '4'
rating '5'
variance
room is comfortable and clean
(room, comfortable) 9 13 15 15 17 10
(is, comfortable) 8 7 9 7 10 2
(comfortable, and) 5 8 11 10 8 4
(comfortable, clean) 8 15 22 23 23 44
highest variance > min_{var}
room is comfortable and clean 8 15 22 23 23
contextual region: $region^{(m)}_{(c_n,d)}$
rating distribution: $dist^{(m)}_{(c_n,d)}$
Train data

Figure 5: Identifying the skip-gram that contributes the highest variance in rating distributions.

"clean") is selected as the label for the region "room is *comfortable* and clean." If no skip-gram exceeds the variance threshold min_{var}, only the candidate context word c_n is chosen as the context for that region, and its rating distribution is chosen as the label. After finding the set of $dist^{(m)}_{(c_n,d)}$ for all $region^{(m)}_{(c_n,d)} \in Region$, they are stored in a rating-distribution matrix $Dist$.

3.1.3 Learning the Region Embeddings for Contextual Regions. We now have the contextual regions *Region* and their associated rating distributions *Dist*. Our objective now is to build a predictive model that, given a contextual region $region^{(m)}_{(c_n,d)}$ as an input, predicts a rating distribution $dist^{(m)}_{(c_n,d)}$ as an output. To achieve this, we need a model that can be used to identify which words in $region^{(m)}_{(c_n,d)}$ contribute to $dist^{(m)}_{(c_n,d)}$. We choose the region embedding with local context unit proposed by [21] as our training model. This model learns the region embeddings, i.e., the representations of the text regions, with the help of word embeddings and local context units. The local context unit is a weighting matrix that captures the interactions between a word and its neighbors in a text region. Our interest here is that the local context unit can be applied to emphasize the positions of the words that have an influence on the rating distributions, and therefore can be considered as part of the context.

Formally, every word w_t has an associated word embedding $\mathbf{e}_{w_t}$, which is stored in the column of the embedding matrix $\mathbf{E} \in \mathbf{R}^{h \times |Vocab|}$, where h is the embedding size and $Vocab$ is the vocabulary of all words in the training data. In addition to the word embeddings, a candidate context word c_n also has its associated local context unit matrix $\mathbf{K}_{c_n} \in \mathbf{R}^{h \times (2 \times d+1)}$, which is stored in the tensor $\mathbf{C} \in \mathbf{R}^{h \times (2 \times d+1) \times |Cand|}$. Note that we only need to learn $\mathbf{K}_{c_n}$ for $|Cand|$ candidate context words, unlike the original model that learned this parameter for all $|Vocab|$ words.

Given a contextual region $region^{(m)}_{(c_n,d)}$ as an input, the projected word embedding $\mathbf{p}_{w_t}$ of word w_t at index position l of $region^{(m)}_{(c_n,d)}$ is calculated by Equation 4.

$$\mathbf{p}_{w_t} = \mathbf{K}_{c_n,l} \odot \mathbf{e}_{w_t} \tag{4}$$

The word embedding $\mathbf{e}_{w_t}$ of word w_t at position l of $region^{(m)}_{(c_n,d)}$ is projected into the point of view of the candidate context word c_n by element-wise multiplying with the corresponding column l of $\mathbf{K}_{c_n}$. This indicates that c_n can alter the semantic meaning of e_{w_t}. For example, w_t = "clean" in the region of c_n = "comfortable" has a different semantic meaning from when it is in the region of c_n = "not."

After obtaining all projected word embeddings, the region embedding $\mathbf{r}^{(m)}_{(c_n,d)} \in \mathbf{R}^{h \times 1}$ of the contextual region $region^{(m)}_{(c_n,d)}$ is computed by applying the max-pooling operation over all projected word embeddings given in Equation 5.

$$\mathbf{r}^{(m)}_{(c_n,d)} = max([\mathbf{p}_{w_{t-d}}\ \mathbf{p}_{w_{t-d+1}}\ \cdots\ \mathbf{p}_{c_n}\ \cdots\ \mathbf{p}_{w_{t+d-1}}\ \mathbf{p}_{w_{t+d}}]) \tag{5}$$

Finally, $\mathbf{r}^{(m)}_{(c_n,d)}$ is fed into the fully connected layer to calculate the rating distribution $dist^{(m)}_{(c_n,d)}$. Although a model was originally proposed for the classification task, we want to predict a rating-distribution vector. We, therefore, adopt a multivariate linear-regression model. Our model can be expressed as in Equation 6.

$$dist^{(m)}_{(c_n,d)} \approx \mathrm{f}(\mathbf{x}; \mathbf{E}, \mathbf{C}, \mathbf{W}, \mathbf{b}) = \mathbf{W} \cdot \mathbf{r}^{(m)}_{(c_n,d)} + \mathbf{b} \tag{6}$$

Here, the parameter $\mathbf{x}$ is an input, i.e., the contextual region $region^{(m)}_{(c_n,d)}$, $\mathbf{W} \in \mathbf{R}^{|rating| \times h}$ and $\mathbf{b} \in \mathbf{R}^{|rating| \times 1}$ are the weight matrix and bias vector, respectively, where $|rating|$ is the size of the categorical rating scores. We chose L2 as our loss function, following [21], and Adam [11] as the optimizer. No regularization is applied.

After all model parameters ($\mathbf{E}, \mathbf{C}, \mathbf{W}$ and $\mathbf{b}$) are learned, each contextual region $region^{(m)}_{(c_n,d)}$ can now be mapped with its region embedding representation $\mathbf{r}^{(m)}_{(c_n,d)}$. Such a region embedding is trained to capture the global influence, i.e., the influence on the rating distribution of the entire review dataset, of its associated contextual region. This means that if two region embeddings are similar in the embedding space, they will be expected to contribute similar rating distributions.

In the next section, we show how the extracted contextual regions and their associated region embeddings can be utilized in the rating prediction task.

3.2 Rating Prediction

The previous section explained how we are able to extract context from reviews as contextual regions and represent them by their associated region embeddings. These region embeddings, however, capture only the *global* influence of the contextual regions on the rating distributions for the entire review dataset.

To make a personalized rating prediction, it is important to model the *local* influence of each contextual region on a particular review provided by an individual user with respect to a specific item. This is because a contextual region might have a range of possible influences on the user's decision about the item that depends on its relevance to the user's personal preferences and the item's unique features. For example, a user might choose a hotel based on the cleanliness of the room but ignore the location of the hotel, meaning that contextual regions containing the word "clean" should have more influence on the ratings for this user than those that contain "location." Similarly, a hotel might be famous for its location, which should have more influence on the ratings it received than its breakfast menu.

To model the local influences of contextual regions, we introduce $\mathbf{T}_u \in \mathbf{R}^{|Users| \times h}$ the *user–context interaction* matrix; and $\mathbf{T}_v \in \mathbf{R}^{|Items| \times h}$ the *item–context interaction* matrix. The parameter h is the embedding size, which is the same as for the region embedding. Each row in $\mathbf{t}_{u_i} \in \mathbf{T}_u$ and $\mathbf{t}_{v_j} \in \mathbf{T}_v$ represent user u_i's and item v_j's interaction vectors, respectively, with the contextual regions. These vectors are learned to capture the past interaction of each user/item with the contextual regions and model the influences of such regions on the ratings for that user/item. For example, if u_i wrote reviews containing the contextual region "room is *comfortable* and clean" with rating "5" a significant number of times, $\mathbf{t}_{u_i}$ will be able to justify this contextual region's relevance to u_i's preferences, because it influences the ratings given by u_i.

The vectors $\mathbf{t}_{u_i}$ and $\mathbf{t}_{v_j}$ can be seen as projection vectors for converting the region embedding $\mathbf{r}^{(m)}_{(c_n,d)}$ (which captures the global influence to the rating distribution of the entire review data) to the region embeddings $\mathbf{r}^{(m)}_{(c_n,d),i}$ and $\mathbf{r}^{(m)}_{(c_n,d),j}$ (which capture the local influences on the ratings of u_i and v_j, respectively). We call $\mathbf{r}^{(m)}_{(c_n,d),i}$ the *user-relevance* region embedding, and $\mathbf{r}^{(m)}_{(c_n,d),j}$ the *item-relevance* region embedding. Specifically, $\mathbf{r}^{(m)}_{(c_n,d),i}$ and $\mathbf{r}^{(m)}_{(c_n,d),j}$ are computed from the interactions between $\mathbf{r}^{(m)}_{(c_n,d)}$ and $\mathbf{t}_{u_i}$ or $\mathbf{t}_{v_j}$ using element-wise multiplication, as expressed in Equation 7.

$$\mathbf{r}^{(m)}_{(c_n,d),i} = \mathbf{t}_{u_i} \odot \mathbf{r}^{(m)}_{(c_n,d)}, \qquad \mathbf{r}^{(m)}_{(c_n,d),j} = \mathbf{t}_{v_j} \odot \mathbf{r}^{(m)}_{(c_n,d)} \tag{7}$$

After the region embeddings for all contextual regions in the review $q_{(i,j)} \in Review$ are converted into user-relevance and item-relevance region embeddings, they are ready to be used to predict the rating of user u_i toward item v_j. Our predictive model is represented by Equation 8.

$$pref_{(i,j)} \approx \mathrm{g}(\mathbf{x}'; \mathbf{T}_u, \mathbf{T}_v, \mathbf{W}', b') = \mathbf{W}' \cdot \boldsymbol{\gamma}_{q_{(i,j)}} + b' \tag{8}$$

This model can be considered as a neural network that uses simple linear regression to predict the rating output $pref_{(i,j)}$, given a review $q_{(i,j)}$ as an input $\mathbf{x}'$. The model utilizes the region embeddings of all contextual regions extracted from $q_{(i,j)}$ to compute $\mathbf{r}^{(m)}_{(c_n,d),i}$ and $\mathbf{r}^{(m)}_{(c_n,d),j}$. The parameter $\boldsymbol{\gamma}_{q_{(i,j)}} \in \mathbf{R}^{h \times 1}$ is the representation of the review $q_{(i,j)}$ in a fully connected layer, which can be computed in various ways depending on the predictive model used. The parameter $\mathbf{W}' \in \mathbf{R}^{1 \times h}$ is a weight vector and $b' \in \mathbf{R}$ is a scalar bias. Similarly to the context extraction model case, we again choose L2 as our loss function and Adam as the optimizer. No regularization is applied.

We propose four different predictive models based on how the user-relevance and item-relevance region embeddings are used. These models are the no relevance (*NR*), user relevance (*UR*), item relevance (*IR*) and user-item relevance (*UIR*) models. They are designed to deal with various cold-start and sparse data situations.

3.2.1 *NR model.*

To show the importance of considering the relevance of context to each user and each item, we first propose a predictive model that ignores the relevance of contexts. In the NR model, the region embeddings of the contextual regions are used to compute $\boldsymbol{\gamma}_{q_{(i,j)}}$ directly without any conversion, as shown by Equation 9.

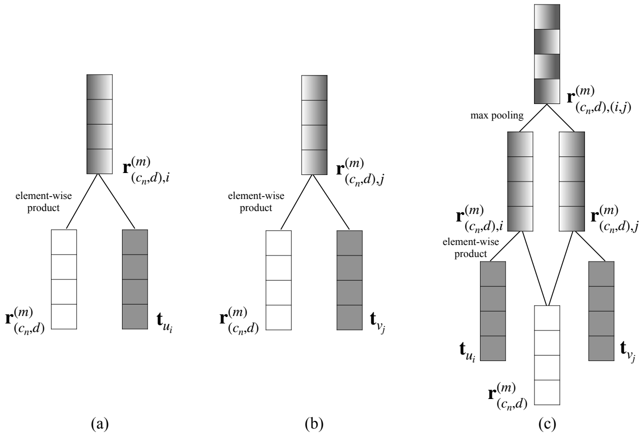

element-wise product
$\mathbf{r}^{(m)}_{(c_n,d),i}$
$\mathbf{r}^{(m)}_{(c_n,d)}$
$\mathbf{t}_{u_i}$
(a)
element-wise product
$\mathbf{r}^{(m)}_{(c_n,d),j}$
$\mathbf{r}^{(m)}_{(c_n,d)}$
$\mathbf{t}_{v_j}$
(b)
$\mathbf{r}^{(m)}_{(c_n,d),(i,j)}$
max pooling
$\mathbf{r}^{(m)}_{(c_n,d),i}$
$\mathbf{r}^{(m)}_{(c_n,d),j}$
element-wise product
$\mathbf{t}_{u_i}$
$\mathbf{t}_{v_j}$
$\mathbf{r}^{(m)}_{(c_n,d)}$
(c)

Figure 6: Illustrations of three of the four proposed models: (a) UR model, (b) IR model and (c) UIR model.

$$\boldsymbol{\gamma}_{q_{(i,j)}} = \sum_{m \in M_{q_{(i,j)}}} \mathbf{r}^{(m)}_{(c_n,d)} \;/\; N \tag{9}$$

Here, $M_{q_{(i,j)}}$ is the set of indexes of $region^{(m)}_{(c_n,d)}$ in *Region*, for all N contextual regions extracted from the review $q_{(i,j)}$. The representation of $q_{(i,j)}$ is computed by averaging all corresponding region embeddings of the contextual regions in $q_{(i,j)}$. Therefore, the predicted rating for $q_{(i,j)}$ does not depend on any interaction with either the user or the item. This means that if two reviews contain exactly the same set of contextual regions, they will receive the same rating. The NR model is suitable for very sparse datasets, where there are few reviews for most users and for most items.

3.2.2 *UR model.*

A graphical representation of the UR model is given in Figure 6 (a). In this model, only the user-relevance region embeddings are used to compute $\boldsymbol{\gamma}_{q_{(i,j)}}$, as given in Equation 10.

$$\boldsymbol{\gamma}_{q_{(i,j)}} = \sum_{m \in M_{q_{(i,j)}}} \mathbf{r}^{(m)}_{(c_n,d),i} \;/\; N \tag{10}$$

The UR model predicts the rating for review $q_{(i,j)}$ considering only the relevance of context to user u_i, with the relevance of context to the item v_j being ignored. The idea behind this model is that the rating a user will give to any item depends only on the suitability of the contextual information to that user's past preferences. With this model, any item with the same set of contexts will receive the same rating from the same user. The UR model is designed for datasets where a cold start is a problem for the items, i.e., most items have only a few reviews.

3.2.3 *IR model.*

Figure 6 (b) represents the IR model. In contrast to the UR model, this model computes $\boldsymbol{\gamma}_{q_{(i,j)}}$ by considering only the item-relevance region embeddings, as expressed by Equation 11.

$$\boldsymbol{\gamma}_{q_{(i,j)}} = \sum_{m \in M_{q_{(i,j)}}} \mathbf{r}^{(m)}_{(c_n,d),j} \; / \; N \tag{11}$$

In contrast to the UR model, the idea behind the IR model is that the rating an item will receive from any user only depends on the suitability of the contextual information to its unique features. This means that any user who chooses the same item with the same set of contexts should generate the same rating. The model is appropriate for datasets where a cold start is a problem for the users, i.e., most users have only a few reviews.

3.2.4 UIR model. The fourth and final model, the UIR model, is slightly more complex than the other three. As shown in Figure 6 (c), the region embeddings $\mathbf{r}^{(m)}_{(c_n,d)}$ are projected to both the user interaction vector $\mathbf{t}_{u_i}$ and the item interaction vector $\mathbf{t}_{v_j}$. Therefore, this model employs both the user-relevance region embedding $\mathbf{r}^{(m)}_{(c_n,d),i}$ and the item-relevance region embedding $\mathbf{r}^{(m)}_{(c_n,d),j}$. We apply the max-pooling operation on these two region embeddings to create the *user-item relevance* region embedding $\mathbf{r}^{(m)}_{(c_n,d),(i,j)}$, as expressed by Equation 12.

$$\mathbf{r}^{(m)}_{(c_n,d),(i,j)} = max([\mathbf{r}^{(m)}_{(c_n,d),i} \;\; \mathbf{r}^{(m)}_{(c_n,d),j}]) \tag{12}$$

The purpose of this combined region embedding is to capture the maximum relevance of $region^{(m)}_{(c_n,d)}$ to the pair of u_i and v_j. For example, this means that, even if a contextual region is not highly relevant to an item's features but it is highly relevant to a user's preferences, it can still affect the rating. For this model, the parameter $\boldsymbol{\gamma}_{q_{(i,j)}}$ is computed by Equation 13.

$$\boldsymbol{\gamma}_{q_{(i,j)}} = \sum_{m \in M_{q_{(i,j)}}} \mathbf{r}^{(m)}_{(c_n,d),(i,j)} \; / \; N \tag{13}$$

Compared with the other three models, this model is more realistic in that contexts should be relevant to both users' preferences and items' features, and should, therefore, influence the review ratings. In other words, the rating that the review will receive depends on the suitability of the contextual information to both the user's preferences and the item's features. The UIR model is suitable for datasets for which there are significant numbers of reviews for both users and items.

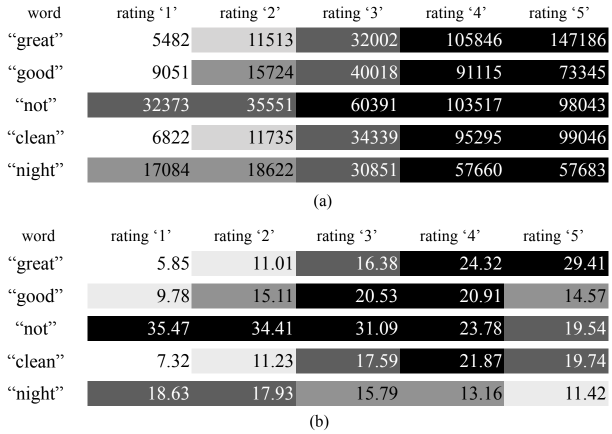
4 EXPERIMENTS AND DISCUSSION

4.1 Data Preparation

We used the publicly available TripAdvisor dataset[1], which contained 878,561 hotel reviews when used for our experiments. We preprocessed the data by tokenizing the reviews and converting all words to lower case. All punctuation marks and infrequently used words (i.e., those of appearance frequency below 0.01% in all reviews) were removed. We also removed all stopwords listed by NLTK[2], except for those that indicate sentiment meanings (e.g., "very" or "not"). After the preprocessing, reviews of less than two words were marked as uninformative and were therefore discarded. The preprocessed-data statistics are given in Table 1. Because the extraction and prediction parts require different types of input, further specific data preparation for each part was necessary.

Table 1: Statistics for the Preprocessed TripAdvisor Dataset

Reviews	873,199
Words per review	Min: 2 Max: 2011 Average: 83.008
Users	575,264
Reviews per user	Min: 1 Max: 63 Average: 1.385
Items	3,941
Reviews per item	Min: 1 Max: 5,426 Average: 221.568
Ratings	1: 53,501 2: 59,711 3: 121,780 4: 291,913 5: 346,294
% Ratings	1: 6.1% 2: 6.8% 3: 13.9% 4: 33.4% 5: 39.7%



word	rating '1'	rating '2'	rating '3'	rating '4'	rating '5'
"great"	5482	11513	32002	105846	147186
"good"	9051	15724	40018	91115	73345
"not"	32373	35551	60391	103517	98043
"clean"	6822	11735	34339	95295	99046
"night"	17084	18622	30851	57660	57683

(a)

word	rating '1'	rating '2'	rating '3'	rating '4'	rating '5'
"great"	5.85	11.01	16.38	24.32	29.41
"good"	9.78	15.11	20.53	20.91	14.57
"not"	35.47	34.41	31.09	23.78	19.54
"clean"	7.32	11.23	17.59	21.87	19.74
"night"	18.63	17.93	15.79	13.16	11.42

(b)

Figure 7: Rating distributions for example words: (a) before standardization, (b) after standardization.

4.1.1 Data for Context Extraction. We randomly chose 90% of the dataset to train the model parameters for the context extraction. Here, the main task was to extract the contextual regions and their associated rating distributions. The first step was to identify the candidate context words that influence the rating distributions. However, the main problem was that some rating datasets contained a bias in the proportion of ratings provided by the users. For example, as given in Table 1, more than 80% of the reviews are rated as "4" or "5." Therefore, almost every word in the corpus was distributed toward high-rating scores, as shown in Figure 7 (a). To make it possible properly to analyze the influence of a word on the rating distribution, we applied a data standardization technique, as expressed by Equation 14.

$$x^{new}_{t,r} = \frac{x_{t,r} - \mu_r}{\sigma_r} \tag{14}$$

Here, $x_{t,r}$ is the frequency of word w_t given for rating r, μ_r is the average of the frequencies of all words given for rating r and σ_r is the standard deviation of the frequencies of all words given for rating r. The rating distributions after applying this standardization are shown in Figure 7 (b). For example, the word "not" is now more distributed toward low ratings, which is more appropriate, given that it indicates a negative meaning.

[1] http://www.cs.cmu.edu/ jiweil/html/hotel-review.html

[2] https://www.nltk.org/nltk_data/

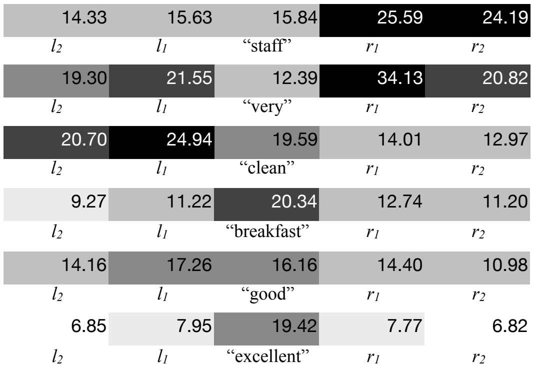
Table 2: Model Parameters for All Evaluated Methods

Method	Rate/User	Rate/Item	Emb. Size	Latent	Learn. Rate
WordEmb	1	1	300	-	0.001
NMF	5	5	100	10	0.001
NMFw2v	5	5	100	10	0.001
NR	1	1	300	-	0.001
UR	10	1	300	-	0.0001
IR	1	200	300	-	0.001
UIR	5	5	300	-	0.0001

After standardization, applying $min_{var} = 1$ gave us a set of 226 candidate context words $Cand$. We set a region size of 5 for the extraction of contextual regions from reviews. Because each $c_n \in Cand$ might be the first or last word in the review, we first added a padding of length $d = 2$ to the head and tail of each review. The result of the extraction gave us 25,077,762 contextual regions. To avoid scalability problems in the training process, we randomly sampled only a subset of the contextual regions for the training. Specifically, let $Region_{c_n}$ denote the set of contextual regions of c_n. If $|Region_{c_n}| > 100$k, only a 10% subset is used for training. If 10k $\le |Region_{c_n}| \le 100$k, 10k are used. If $|Region_{c_n}| < 10$k, all are used. After this process, 3,125,212 contextual regions were selected as the training regions. We generated skip-grams of size $\rho = 2$ and assigned the rating distribution with the highest variance to each region, using $min_{var} = 1$.

4.1.2 Data for Rating Prediction. We evaluated the predictive performance of the four models using a fivefold cross-validation technique. The model parameters for our predictive models and for the other methods evaluated are presented in Table 2. We compared our proposed models with three other methods, namely word embedding for regression (WordEmb), nonnegative matrix factorization (NMF) and NMFw2v. WordEmb is the method whereby word embeddings are learned to predict the rating for each review (i.e., we compute the average of all word embeddings in a review, and feed it to the fully connected layer for prediction). NMFw2v [18] is a version of NMF, extended to incorporate the user and item embedding profiles, as expressed by Equation 2. We followed [18] to set values for the latent dimension and embedding sizes and used Gensim Word2Vec[3] to learn the word embeddings for all words in our corpus.

To train the NMF, NMFw2v, UR, IR and UIR models, we selected only users and/or items with a significant number of reviews, to ensure that these models have sufficient data to learn the embedding profiles and/or the interactions with contexts. For example, only reviews from users who had provided more than 10 reviews were used to train our UR model, thereby producing high-quality user-context interaction vectors. For our predictive models, all reviews with no candidate context words were discarded, which resulted in different sizes of training and test datasets for each model. Finally, we trained all models using the Adam optimizer, with L2 as loss function. The regularization parameters for NMF and NMFw2v were set to 0.1 and 0.001, respectively.



14.33	15.63	15.84	25.59	24.19
l_2	l_1	"staff"	r_1	r_2
19.30	21.55	12.39	34.13	20.82
l_2	l_1	"very"	r_1	r_2
20.70	24.94	19.59	14.01	12.97
l_2	l_1	"clean"	r_1	r_2
9.27	11.22	20.34	12.74	11.20
l_2	l_1	"breakfast"	r_1	r_2
14.16	17.26	16.16	14.40	10.98
l_2	l_1	"good"	r_1	r_2
6.85	7.95	19.42	7.77	6.82
l_2	l_1	"excellent"	r_1	r_2

Figure 8: Visualization of the local context units for some chosen candidate context words.

4.2 Results and Discussion

In this section, we first visualize and discuss the extracted candidate context words, together with the influence of their neighboring words on the rating distributions. We then present the predictive performances of our predictive models compared with the other methods and discuss these results.

4.2.1 Influences of Candidate Context Words and Their Neighboring Words. As discussed in Section 3.1.2, their neighboring words might alter the influence of candidate context words on their rating distributions. To analyze such influences, we follow [21] by applying the L2-norm to each column of the local context unit. This enables the influence levels of the candidate context and their neighboring words to be emphasized, as shown in Figure 8. For example, words that follow "staff" and "very" have more influence on rating distributions than the words that come before them. This corresponds to the following words often being "good," "helpful" or "friendly" for "staff," and "clean," "convenient" or "comfortable" for "very." On the other hand, words such as "breakfast" are less influenced by neighboring words, meaning that the word itself sufficiently describes the rating distributions without any help from neighboring words. Moreover, the local context units can differentiate the influence of positive words such as "good" or "excellent." Although the rating distributions of "good" are influenced by its neighboring words, the word "excellent" is not. This is because the word "excellent" itself indicates the strongest positive meaning, whereas the semantic meaning of "good" can be altered if it follows words such as "not" or "very."

For these reasons, we see that the local context units can capture the influences of the candidate context words efficiently, together with their neighboring words, on the rating distributions. This further helps to produce high-quality region embeddings, which are capable of semantically representing the distribution of ratings for the individual contextual regions.

4.2.2 Predictive Performances. The prediction accuracy (MAE and RMSE) of our predictive models, compared with the other methods, are presented in Table 3. From the table, our models provide the best prediction accuracy, followed by WordEmb, NMFw2v and NMF.

[3] https://code.google.com/archive/p/word2vec/

As compared with WordEmb, which exploits embeddings of all words to predict a rating, our NR model yields a similar prediction accuracy. This means that exploiting a smaller number of embeddings from the contextual regions is sufficient to produce an effective predictive model. The accuracy, however, can be further improved by considering the relevance of contexts to user and/or item, as shown by the results from the UR, IR and UIR models.

As compared with NMF, all other methods that incorporate the review data provide better results. This means that the textual content help improves the prediction accuracy over the model that considers only the preference data. However, the NMFw2v model, which exploits the pretrained word embeddings to create user and item profiles, produces no significant difference in accuracy compared with the baseline NMF. In our experiment, we are able to get the best prediction accuracy for NMFw2v with $\alpha = 0.95$. This means that NMF contributes to most of the prediction, not the user or item embedding profiles. If we compare this method with WordEmb, which learns word embeddings directly for the prediction task, NMFw2v yields a lower accuracy. In our case, we might assume that learning the word embeddings directly for prediction is more efficient than exploiting the pretrained word embeddings.

Now, let us compare the predictive performances between our four predictive models. First, the UR, IR and UIR models outperform the NR model, meaning that considering the relevance of contexts to user and/or items improves the accuracy of the model. However, because the NR model does not depend on the past interactions between the users or items with contexts, it can make a prediction even if a user or item has no review data. Similarly, the UR model does not require any item's interaction, while the IR model does not require any user's interaction with contexts. These three models are suitable for dealing with different cold-start scenarios, where there are new users and/or items with no past interaction with the systems. For the UIR model, it provides the best overall accuracy among all other models because it considers the relevance of contexts to both users and items, and models their local influences to the rating. However, its accuracy comes with the trade-off with the prediction coverage because it requires a significant number of reviews from both users and items to model their interactions with the contextual regions. We, therefore, conclude that our four predictive models are suitable for different situations, depending on the characteristics of the review data. If the review data are dense, both users and items have a significant number of reviews, the UIR model is recommended. If the data has user or item cold-start problems, then UR or IR is preferable. Finally, though less accurate, the NR model is more robust to the sparse data because it does not require any interaction from users or items.

Table 3: Prediction Accuracy Results

Method	MAE	RMSE
WordEmb	0.8962	1.1653
NMF	1.3530	1.6478
NMFw2v	1.1377	1.8809
NR	0.9038	1.1661
UR	0.7623	0.9806
IR	0.8371	1.1169
UIR	**0.7400**	**0.9647**

5 CONCLUSION

We propose a novel unsupervised context extraction method for review data, along with the predictive models for rating prediction. Because our method automatically extracts contexts from reviews, there is no need to predefine the optimal values of contexts. Unlike any previous context-aware method, our context is defined in a form of skip-gram, meaning that it can be constructed from any word in the same text region. This makes our method suitable for extracting contexts from reviews, where users have different written styles to indicate their contextual information. Moreover, we consider the influences of contexts to both population and individual levels. While the influences of contexts on the entire reviews' ratings are captured by their region embedding representations, the rating prediction is made by considering the relevance of those contexts to the individual users and items. The four predictive models make our proposed method suitable for dealing with different cold-start and sparse data situations. Experimental results show that our models yield better prediction accuracy than the review-based recommendations that do not consider contexts.

ACKNOWLEDGMENTS

This work was supported by "Cross-ministerial Strategic Innovation Promotion Program (SIP) Second Phase, Big-data and AI-enabled Cyberspace Technologies" by New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. 1999. Towards a better understanding of context and context-awareness. In *Handheld and Ubiquitous Computing*. Springer Berlin Heidelberg, 304–307.

[2] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems* (2005), 103–145.

[3] Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. 2012. Context relevance assessment and exploitation in mobile recommender systems. In *Personal and Ubiquitous Computing*. 507–526.

[4] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. 2011. Matrix factorization techniques for context aware recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*. ACM, Chicago, USA, 301–304.

[5] Konstantin Bauman and Alexander Tuzhilin. 2014. Discovering contextual information from user reviews for recommendation purposes. In *1st Workshop on New Trends in Content-based Recommender Systems, co-located with the 8th ACM Conference on Recommender Systems, CBRecSys@RecSys 2014*. ACM, Silicon Valley, USA.

[6] Guanliang Chen and Li Chen. 2014. Recommendation based on contextual opinions. In *User Modeling, Adaptation, and Personalization*. Springer International Publishing, 61–73.

[7] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction* (2015), 99–154.

[8] Paul Dourish. 2004. What we talk about when we talk about context. *Personal and Ubiquitous Computing* (2004), 19–30.

[9] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2012. Context-aware music recommendation based on latent topic sequential patterns. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. ACM, Dublin, Ireland, 131–138.

[10] Negar Hariri, Bamshad Mobasher, Robin Burke, and Yong Zheng. 2010. Context-aware recommendation based on review mining. In *Proceedings of the 9th International Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP 2011)*. Barcelona, Spain.

[11] Diederik Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *3rd International Conference on Learning Representations* (2014).

[12] Yize Li, Jiazhong Nie, Yi Zhang, Bingqing Wang, Baoshi Yan, and Fuliang Weng. 2010. Contextual recommendation based on text mining. *proceedings of the 23rd International Conference on computational Linguistics: Posters* (2010), 692–700.

[13] Yuchen Li, Dongxiang Zhang, Ziquan Lan, and Kian Lee Tan. 2016. Context-aware advertisement recommendation for high-speed social news feeding. In *2016 IEEE 32nd International Conference on Data Engineering, ICDE 2016*. Helsinki, Finland, 505–516.

[14] Liwei Liu, Freddy Lecue, Nikolay Mehandjiev, and Ling Xu. 2010. Using context similarity for service recommendation. In *Proceedings of 2010 IEEE 4th International Conference on Semantic Computing, ICSC 2010*. IEEE Computer Society, Washington, DC, USA, 277–284.

[15] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. Matrix factorisation with word embeddings for rating prediction on location-based social networks. In *Advances in Information Retrieval, ECIR 2017*. Springer International Publishing, Aberdeen, Scotland, 647–654.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13)*. 3111–3119.

[17] Cataldo Musto, Giovanni Semeraro, Marco De Gemmis, and Pasquale Lops. 2015. Word embedding techniques for content-based recommender systems: an empirical evaluation. In *CEUR Workshop Proceedings*. Vienna, Austria.

[18] Ante Odić, Marko Tkalčič, Jurij F. Tasič, and Andrej Košir. 2013. Predicting and detecting the relevant contextual information in a movie-recommender system. *Interacting with Computers* 25 (2013), 74–90.

[19] Francisco J. Peña. 2017. Unsupervised context-driven recommendations based on user reviews. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, 426–430.

[20] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, 1532–1543.

[21] Chao Qiao, Bo Huang, Guocheng Niu, Daren Li, Daxiang Dong, Wei He, Dianhai Yu, and Hua Wu. 2018. A new method of region embedding for text classification. *International Conference on Learning Representations* (2018).

[22] Padipat Sitkrongwong, Saranya Maneeroj, Pannawit Samatthiyadikun, and Atsuhiro Takasu. 2015. Bayesian probabilistic model for context-aware recommendations. In *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services (iiWAS '15)*. ACM, Brussels, Belgium.

[23] Moshe Unger. 2015. Latent context-aware recommender systems. *RecSys 2015 - Proceedings of the 9th ACM Conference on Recommender Systems* (2015), 383–386.

Privacy in the Age Of Information (and algorithms)

Bipin C. Desai[*]
BipinC.Desai@concordia.ca
IDEAS/Concordia University
Montreal, Canada

ABSTRACT

This paper raises the privacy issues related to information that is accessible about individuals from their mobile devices and that which is collected when they interact with and use so called "free" services provided on the web. The importance of privacy has been ignored by most legislation and any laws passed have no teeth. The only exception is the privacy protection that is embedded in the EU's General Data Protection Regulation(GDPR). GDPR gives control to individuals over their personal data and requires any organization which collects and controls personal information to have in place appropriate measures both technical and logistic, to implement the data protection principles. In this paper, we propose a technical solution to provide a personal email and web server with complete control of all correspondence and contents. This would liberate users from fake free services and provide privacy and security.

CCS CONCEPTS

• **Security and privacy** → **Firewalls**; **Privacy protections**; • **Networks** → **Middle boxes / network appliances**; • **General and reference** → *Experimentation.*

KEYWORDS

Privacy, security, online social networks, free email service, warrantless constant surveillance, Heimdaller

ACM Reference Format:
Bipin C. Desai. 2019. Privacy in the Age Of Information (and algorithms). In *Proceedings of 23rd International Database Engineering & Applications Symposium(IDEAS'19) (IDEAS 2019).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.475/3331076.3331089

1 PRIVACY ITS RISE AND FALL

According to some, privacy was the result of the growth of the middle class who could afford better housing and their abode, even though humble, became their castle. Liberalization of laws such as the one that took the state out of the bedrooms recognized the right of people to be left alone. However, this was only at the governmental level since it had supreme power but the corporations were left to regulate themselves. At the end of the first gilded age, the need was felt to regulate the robber barons and their corporation..This was done by the emergence of the workers unions and the need to share the riches with the worker. The later was implemented with a progressive tax system that supported the common good by taxing the haves to transfer to the have-nots. This afforded the luxury of privacy to the not so rich! It must be noted that recently, the Supreme Court of the largest functioning democracy in the history of humankind has ruled that privacy is a human right [44]: this in spite of an inane pronouncement of a kid who became fabulously rich and influential[53].

The tools and technology that are the roots of the privacy exploitation required many developments quite a number of these occurred in the mid 20th century. The first of these was the introduction of computers and the development of semiconductors, its miniaturization and increase in computing power. The interconnection of computers and hence people was made possible with the introduction of the Internet. In the early days, access to the internet was limited to the academic communities, some businesses and government agencies. The introduction of the web and the development of the graphical browsers opened up the internet to an increasing number of users.

The offer of web based free email service by start ups fueled by venture capitalist allowed these companies the continuous access to all email communications of an increasing number of users world wide! The undeclared charge for this free service was and continues to be the contents of their messages and these users who in turn attracted more users and more contents. There being no laws to protect the privacy of the contents of these messages, which are in plain text. This clear text being the raw material for tailoring targeted publicity to the users of this free email service was the raw material used initially for monetizing the 'free' service.

The graphical web browser also opened up a new method of communicating with family and friends. However, since the majority of users were seduced by the allure of setting up a fee web presence without the need to be tech-savvy, this was the start of what we now call online social networks(OSN). The OSN attracted not only friends and family but complete strangers. This attraction of becoming a celebrity has been the force that pushed up the number of such users to billions. To this date, instead of recognizing themselves as publishers, these OSN claim to be simple platforms and take no responsibility for the content they provide which has created many problems, including bullying, extreme self-harm, fake news, genocide and terrorist act co-ordination[95].

The web opened up the internet to the non-tech savvy user. Instead of creating tools to make the Internet service such as email to be used privately, securely and easily accessible to the non-tech savvy or offering it via the national postal service, the unimaginative politicians let it be provided by venture capitalists.

[*] Corresponding Author

IDEAS 2019, June 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.475/3331076.3331089

A system such as the web robot and allowing any robot to be served freely by the web servers was one of the biggest blunders made by the web community. The taking over of the web by the commercial players and facilitating tracking in the web browser are other blunders driven by monetizing the web. It is reported that there is an attempt to undo the harm in the new web but again, some of the players are the same as in the first web, and the same commercial pressure would jeopardize this solution.

The cell phone which started out as a bulky device profited from the miniaturization of semiconductor circuits and started getting smaller, lighter and more popular. The addition of a screen and more computing power and better and better networks transformed the cell phone from an emergency device to a personal communication device. The tech giant companies made sure that the new 'smart' cell phones have access to their services and the cell phone replaced the personal computer and the land line. The cell phone system also allowed bypassing the need of setting up an extensive telecommunication infrastructure and the need for laying and maintaining cabling and relay stations. This was replaced by cell phone towers and relay stations. The mobile system speeded up communication in emerging economies as well as the developed ones.

With the growth of the internet and the mobile communication infrastructure, the opportunity of gathering data on individuals from their communications and interactions became relatively easy. The set up of the Global Position System in the early seventies by the USAian government [46] along with the worldwide free access to the Standard Positioning Service (SPS) provided precise location information. System to use of the GPS location is built into the current generation of mobile hand sets. The many applications available for the mobile phone made it possible to introduce services many of which require SPS. The use of the global positioning system allowed the various applications running on cell phones to keep abreast of the location of the user. Some of these, useful to the user for applications such as directions and maps, allowed the marketing of nearby businesses to the cell phone user.

Furthermore, the applications on the mobile system allowed their developer to precisely know the whereabouts of the mobile device and hence its owner. These locations are recorded by the application developer and the supplier of the mobile operating system. Along with the web and the cell technology, the access to users data in their communications and by tracking their use of the web and applying the advances in computer science including data management and algorithms for machine learning etc. created what Zuboff calls Surveillance Capitalism[99]. The exploitation of personal data by private corporations is finally drawing the attention of scholars, and columnists and it is finally reaching the masses. The addiction to the cell phone means that one is constantly looking at it even when out in company with friends and family for a meal or just walking around.

The recognization of corporations as a legal entity made it possible to put in power politicians which reversed the progressive nature of the taxation in the wrong belief that there would be a trickled down effect from the haves to the have-nots. Unfortunately, this effect has not occurred and the growth of the middle class has been halted and reversed. Competition by having another company provide similar, privacy oriented service, seems hardly possible. Because of the large share of the market another similar OSN, even one such as Google+ did not succeed.

1.1 Exposure to Privacy in the Computer Science Curriculum

Many of the current tech giants are headed by people who may have followeda computer science program and/or "tech-geeks" some of them being drop outs. One can safely assume that the majority of the coders could have had a computer science related education. However their exposure to humanities and social sciences would have been very limited if null as it is in many CS programs. The curriculum recommendation from ACM/IEEE includes the following: "A computer engineering curriculum must include preparation for professional practice as an integral component. These practices encompass a wide range of activities including management, ethics and values, written and oral communication, working as part of a team, and remaining current in a rapidly changing discipline." However not much is said about issues of privacy and security except if it is not ethical. However, with the recklessness shown by the robber barons of the late 20th - early 21st century who go ahead like bulls in a china shop and seem to have no regards for a person's privacy.

The sample curriculum for Computer science which runs into hundreds of pages[3] includes exposure of the student to Social Issues and Professional Practice and the documents point out that "Graduates should recognize the social, legal, ethical, and cultural issues inherent in the discipline of computing. They must further recognize that social, legal, and ethical standards vary internationally. They should be knowledgeable about the interplay of ethical issues, technical problems, and aesthetic values that play an important part in the development of computing systems. Practitioners must understand their individual and collective responsibility and the possible consequences of failure. They must understand their own limitations as well as the limitations of their tools." In the section SP/Privacy and Civil Liberties which is 2 Core-Tier1 hours where philosophical, legal, privacy tools and implication and related issues are presented. This hardly seems adequate and is likely to run off the proverbial duck's back. Since some of the aspects of this responsibility is not encouraged to be practiced in the rest of the program the final impact is almost nil. What is puzzling is that in Appendix C of this document where course exemplars are given, there is not a single one just for SP/Privacy and Civil Liberties. One, given on page 304 on Social Issues and Professional Practice, is part of a course which includes Human Computer Interaction and Graphics and Visualization. Another example is Ethics & the Information Age [3], (p436) which however does not touch on the philosophical issue of property, person-hood and the right of a person to privacy. In Stanford university's CS program the course CS181 – Computers, Ethics, and Public Policy allocates a scant 1.6 hours to Privacy & Civil Liberties [3] (p501).

The privacy and security framework[19] of the Canadian Institute for Health Information (CIHI), an independent, not-for-profit organization provides essential information on Canada's health systems and the health of Canadians. Most engineers and software designers are not very well exposed to privacy and may have been exposed minimally to security. However they and the marketing people would likely ignore most of the issues in such frameworks.

The privacy and security page of the USAian Federal Trade Commission(FTC) has the following about data security: “Many companies keep sensitive personal information about customers or employees in their files or on their network. Having a sound security plan in place to collect only what you need, keep it safe, and dispose of it securely can help you meet your legal obligations to protect that sensitive data. The FTC has free resources for businesses of any size”[39]. The guidelines are only for self regulation and the penalty is fairly small; as reported recently, about 22 million USD[35]. The issue addressed by FTC is based on the agreement it had with Facebook for privacy but the FTC claims that the company “deceived consumers by telling them they could keep their information on Facebook private, and then repeatedly allowed it to be shared and made public” [35]. Compare this paltry sum with the one in the guidelines for the EU which call for maximum penalties or 20 million Euro or up to 4% of the world wide revenue for a single breach which can add up to billions of Euros[7]. In the USAian system the privacy issues are being handled by a trade commission, not a human rights agency.

Even though there is so much concern about security, there have been some large breaches in the recent years. Many systems store sensitive information such as passwords in clear text. The fact that the tech giants share information with third parties is enough for one to opt out of any system that needs third parties to carry on their central tasks.

2 SOURCE OF PRIVACY VIOLATIONS

The biggest sources of privacy violation are invisible. On-line shopping requires passing valuable personal information to big as well as small retailers. Some of them are fly-by-night ones while others are multi-billion dollar enterprises. Many small fries are on the coattails of specialized shopping portals. Many of these retailers to increase their revenues, turn around and sell the personal information to data aggregators. The portals also could have access to such data and can use it to direct publicity for products and service to the users and with use of cookies and trackers all this data goes into many different data repositoriesto be exploited, ad infinitum.

While shopping or doing any operation on line„ one is tracked by a myriad of trackers. A case in point is a session with one’s own bank. If one has a tracker reporting add-on in the browser, e.g., Privacy Badger, one see the trackers used by these banks to track their own costumers and share the data with these third parties! A question sent to the bank of why this is being done is never responded to!

2.1 A Typical privacy agreement

When a person signs up as a user of most of the ‘free’ on-line services or to services such as mobile phone supplier, she accepts, unread much less with a clear understanding of what it implies, their privacy policy which is linked to equally unread and not understood data policy. These policies may be updated without the users’ consent. As an example if one considers the privacy policy [34] and the data policy [33] of Facebook, which runs to, in the version currently accessible, 7 and 9 pages respectively; it is no wonder no one reads these and assumes that these privacy policies mean that the site will keep her personal information private and would not share it without her permission[84]. Little does the unsuspecting person knows that she is giving away a free license to persons and organizers who believe that privacy is no longer a social norm[53].

The user is required to let the supplier of the service reserve the right to process, sell, trade or rent aggregated or the users information which is anonymized. As it is well known by now that most anonymizing scheme can be thwarted by combining information from multiple sources. The information that is up for grab includes[1]

Personal information: including name, mailing(postal) address, email address, telephone number, IDs of accounts, device identifiers, PIN, service provider information, account including credit card credentials, passwords, records of all communication as well as details of contacts.

Applications: All providers of applications have access to not only their own application data but also may share this data with other applications on the device. This looks like a modus operandi of all application developers and as Zuboff says, anything that is not guarded would be claimed by these new pirates.

Back-up data on cloud: Would have access to users’ personal information including contacts, email addresses, calendar, memo, tasks, display pictures, status messages, photos, audio, videos – the stated reason is to be able to restore this information.

Cookies: These were introduced in the web space to overcome the stateless nature of the web protocol. The reason for stateless nature of web was due to the philosophy of free sharing of knowledge. However, cookies and their derivatives have morphed into a nefarious form to facilitate surveillance.

Financial Information: Any transaction through the system may require credit status checking etc. any or all of which could be recorded and shared with other parties.

Third party information: The service provider may combine your information with ones obtained from other sources.

Retention Personal information:, Even after the expiry of any direct association with the service provided it could be retained perhaps in an anonimized form and may be used perpetually.

International operations and onward transfers: The service provider, would require you to consent that your personal information may be collected, used, processed, transferred or stored in multiple jurisdictions.

Communication: The service provider may communicate information, surveys, marketing materials, advertisements or personalized content. The service provider may share your personal information within the service provider and with their service providers, financial, insurance, legal, accounting or other advisors.

Here are some of the things these systems have your permission to lay claim on! Any information and content you provide or they collect from creating or sharing content, contents of messages or communications with others. and all information provided while using any of their products including information of the account. They collect details about your connections, address books. logs,

[1] The following is based on the privacy/data agreement of a number of organization including - Apple, Blackberry, Google, Facebook, etc.

meta-data and contents of all communications including all SMSs and emails; pattern of usage including what, when, where, who (and use their algorithms to try to figure out why!). All transactions made which includes purchases which would include the details of the credit/debit cards used, authentication information, addresses and contact information about the transaction. In addition they have access to actions taken by your contact and the information they provide. Your location information is used to determine where you live, where you go, what events you attend and where you are at any point in time. All this information is used to create targeted publicity which is tailored to influence you, using your foibles determined by their unknown algorithms.

The proliferation of the internet via the medium of the web to offer all types of services requires a user to sign-up using a user name and a password. Since more and more services, such as news, financial. Governmental. social and commercial are now offered through the web a typical user may have scores of user IDs and passwords. The tech giants, to increase their presence, have offered to entered into an agreement with many of these services to let the users employ these tech giants credentials to log into these services. Thus the tech giants can trace the user not only on their own platform but can have access to what other services are being used and whatever other information the target service may provide the tech giant. What and how the information these giants would glean besides associating yet another data point in the profiles for these users is not advertised or communicated to the user.

3 PRIVACY VIOLATION AT ANY LEVEL OF SHARING

One of the culprits in the current loss of privacy is the USAian system, its constitution and the outlook of its capitalistic system. Whereas there are some forms of restraint for the USAian governments collecting and using personal information in its constitution and amendments, the private sector is left alone to do as it pleases with a laissez faire self policing attitude. What the citizens do not trust the government to spy on is allowed to the private corporations. That self regulation does not work is amply illustrated in the recent Boeing 737 Max's design flaws which led to two deadly crashes. An optional display that showed the disagreement of the angle of attact sensors on the Boeing Max required additional cost to the millions for the plane.

Furthermore, the fact that Boeing was able to get away with not having the Federal Aviation Agency(FAA) really act as an independent quality control shows that self regulation is unrelaible. According to [31], [28]. "The problems were apparently compounded by FAA rules allowing manufacturers to essentially self-certify aircraft. Boeing reportedly tried to speed up the process in order to catch its rival Airbus A320neo, and pushed the FAA to give it more responsibility. There wasn't a complete and proper review of the documents," a former Boeing engineer said. "[The] review was rushed to reach certain certification dates." [31]. The failure of the correct software and the required equipment for a high priced air-frame leads one to conjecture the type of security employed by many of these tech giants who have no regulation, no oversight and no competition and pay little taxes. They fail to reveal the breach of security or the lack of it for months and years. According to the press, Google did not reveal a security breach for fear of regulations. [29]

With current internet and wireless technology, people actually pay to use the free services in the form of internet connection monthly charges, buy and pay the connection fees for 'smart' devices that allow them to be tracked. Unlike criminals who are tracked by a tracking device imposed on them most consumers now carry a tracking device and pay for it handsomely, every month including for the bandwidth used for tracking.

The result of the USAian system, where the tech giants are based, is that the private sector has laid claim on personal and private information of the users of the myriad of devices that they own. Most of the smart devices are controlled by just two operating platforms again controlled by USAian tech giants. In addition, they control the application stores that users can download the 'apps' from and earn a percent of the fees for these applications. One wonders if this is not an example of a monopoly! Example of such laying a stake, like the one used in the gold rush of yore, is to claim all human experience as free raw material without any concern for individual rights and without any payment of any source[99]. As Zuboff compares these to the edict recited by the Spanish conquistadors and later the settlers of the west in what is now known as the U. S. A. This edict gave the conquistadors and the settlers some form of divine rights which allowed them to usurp the lands of the existing people and displaced them or wiped them out[99].

Google made six cooked up declarations which confer on themselves the right to translate the recorded experience of its users into behavioral data and own it, abuse, use and share it as they see fit and preserve these for perpetuity. They had no problem getting all this data since they had captured the search, the email, the the cell phone markets. They also were in control of the application market place for their cellphones. Another instance of conquest by declaration is the self proclaimed one by the Facebook founder which stated that privacy as no longer a social norm. This statement from a person with very little background in privacy was convenient since it was the basis of Facebook's business model[53] and this declaration, along with a changeable data/privacy policy has been used to mine the information entrusted to them by unsuspecting users. Facebook's usage of this data has been seen to violate the users privacy in many ways. This includes influencing them not only to buy products and services of questionable need but also to expose them to fake and biased news and help create targeted persuasive ads to influence a vote for doubtful candidates and proposals. It is no wonder, over the years Facebook has faced increasing scrutiny borne out by the number of times it has been cited by the privacy commissions, the courts and the popular press[30]. Facebook allowed phone company [58] and other tech giants access to access user data.[27]: they stretch and overstep privacy and competition laws and should be regulated urgently[58]. Others have [23] and want to take Facebook to court[45].

According to the summary of the final report[72] of UK's Digital, Culture, Media and Sport Committee: "among the countless innocuous postings of celebrations and holiday snaps, some malicious forces use Facebook to threaten and harass others, to publish revenge porn, to disseminate hate speech and propaganda of all kinds, and to influence elections and democratic processes—much of which Facebook, and other social media companies, are either unable or unwilling to prevent. . . . ..The big tech companies must not be allowed to expand exponentially, without constraint or proper regulatory oversight. But only governments and the law are powerful enough to contain them. The legislative tools already exist. They must now be applied to digital activity, using tools such as privacy laws, data protection legislation, antitrust and competition law. If companies become monopolies they can be broken up, in whatever sector. Facebook's handling of personal data, and its use for political campaigns, are prime and legitimate areas for inspection by regulators, and it should not be able to evade all editorial responsibility for the content shared by its users across its platforms."[89] Even the people who were involved in the early days of Facebook and its mentor seem to agree with the findings of this and other reports. [66], [48] After having collected millions of email addresses, Facebook says they would stop this practice and notify users[47].

Facebook has used parental influence to mold UE laws[40] and put pressure on politicians, around the world, by promising local investment such as installing data centers in exchange for lobbying for the company to block privacy laws and any forthcoming laws should be Facebook friendly[16], [90]. The fact that the earnings the companies make by their presence in a country is not being taxed is something that the tech giants have been successful in protecting and they continue to lobby for it[90]. Facebook allows governments to target individuals and groups to the extremes, e.g., Rohinga genocide[51], [55] The new virage of Facebook to privacy seems to be fake and meant to decrease their civil liabilities and in fact yet another business spin to try to protect their dominant position and keep at bay the regulations and corporate breakup[13] [88]. Some demands for investigating the lobbying of tech giants are ignored by those in power who hope to benefit from their largess at the election time[71].

3.1 Examples of privacy violations

Over the years, there have been many instances of violation of the common notion of privacy. Even the blanket surrender of privacy in the privacy agreements of the tech, giants is often not honoured, much less the notion of privacy formed over the last few centuries. An overall view is recently reported in [93] that Google's street view violates privacy by taking videos of private homes spaces along with people therein and publishes without any authority. When met with resistance, held off and returned when no one was looking.

Facebook Beacon published purchases made by users without their express consent. Facebook uploaded email contacts of 1.5m users without consent and when discovered says it was inadvertent. Actually it used a feature of a previous version. As usual the information mined from the user contacts and propagated into other databases may not be deleted but used. More of deny, deflect etc.

Google says a microphone in one of their products, which was not revelaed to the buyers, was never activated; one has to take this with a grain of salt when the courts have to tell them to take down world-wide, search results of selling on the web products manufactured in violation of trade secrets [18] There have been many instances of tech companies being warned about privacy. One such is the report by Denham the Assistant Privacy Commissioner of Canada[26]. At that early date the report concludes "that Facebook did not have "safeguards in place to prevent unauthorized access by application developers to users' personal information, and furthermore was not doing enough to ensure that meaningful consent was obtained from individuals for the disclosure of their personal information to application developers"[26].

There is the class action suit against Facebook that is going on for years in British Columbia and the company has used all its resources to keep this from being resolved. The case concerns the practice used by Facebook as of 2011 to feature, users' 'likes' in publicity without the explicit users' consent. The class action was filed in May 2014[23]. The company denied it saying that the consent was automatic and fought it all the way to the Supreme Court of Canada and after many years, the case was won by the plaintiff and the class action was returned to the BC courts after close to four years. It may take a few more years before the class action suit is decided and of course there would be appeals and likely trips back to the supreme court. In the meantime most people would give up and this is what companies with deep pockets able to hire the best lawyers count on. For not obtaining explicit consent from users to use their data, Facebook is facing a fine of up to 5 billion USD from the USAians Federal Trade Commission.

Companies claim that they protect your data; however, it seems that in fact they exploit it and being hacked as reported in the popular press time and again. The number of breaches of data from companies is affecting more and more people since the early days when Apple stored passwords in the clear and had to grudgingly own up[93] to it

3.2 Childrens' Privacy

Children's Online Privacy Protection Act (COPPA) [38] this two decades old USAian federal act protects childrens privacy by giving parents tools to control what information is collected from their children online. The personal information consists of: a first and last name; a home or other physical address including street name and name of a city or town; an e-mail address; telephone number; a Social Security number; any other identifier that could determine the physical or online contacting of a specific individual; or information concerning the child or the parents of that child that the website collects online from the child and combines with other identifiers . A number of tech giants have been fined under the COPPA violation. TikTok is a OSN for video-sharing application and it is alleged to not seek parental consent before collecting information from children under 13 years old[87]. The company is by the governments in India and Bangladesh and has been fined in the USA[86].

Other on-line tech giants let children run up credit card charges using in application charges while playing games on devices such as iPad and iPhone. This kind of preying on children has been going on for a long time as illustated in a story involving Farmville, a Facebook game, reported in 2010[52].

3.3 Legal Actions

The Privacy Commissioner of Canada had launched an investigation in 2018 to examine if Facebook's practices are in compliance with

Canada's federal private sector privacy law, the Personal Information Protection and Electronic Documents Act called PIPEDA)[69] However, this was not the first time: There were early warning in 2009 about the privacy issues with OSN such as Facebook [63], [83]. Many of the complaints found Facebook to be in contravention of the Act and Facebook was to take corrective measures. However, as in the class action launched by Deborah Douez, the case has been going on over many years and is yet another example of the deny, deflect and defend mentality of these tech giants[23], [24], [25]. In a more recent report of joint investigation of Facebook by the Privacy Commissioner of Canada and the Information and Privacy Commissioner for British Columbia the conclusions drawn are that Facebook failed to obtain valid and meaningful consent of users nor their friends. Furthermore the company did not have adequate safeguard to protect users information and was not accountable for the information under its control[73]. The selective restriction used by Google for example in Google v Equustek Inc, was found to be not sufficient and the request for a world wide ban was upheld.

The availability of free widely used OSN platforms allows any one to post anything on it. The posters range from ignorant and zealot bigots, paid geeks, agents of governments to misinformed twitters. After many denials and deflections some of these OSN are finally admitting that their platform is a vehicle for fake news etc. [43] and making a feeble attempt to do something. Where the attempt is lack-luster, a mere 40 people, to fight millions of potential sources of fake news. The company is making sure to get as much spin out of it as possible by inviting dozens of journalists into the 'war' room to fight this fake news; there being a claim that these crews are backed by other unnamed and unseen experts and of course the unknown, unproven algorithms!

While these tech giants claim to be not evil and want people to connect, they are in fact exploiting the recorded human experience to enrich themselves. By using the leverage of different kinds of equity(more than one vote for some types of shares, no vote for others or/and and not allowing some of the voting shareholders to vote against members of the board), they retain the majority voting rights and make sure that the reins of these tech giants are preserved in a dynastic fashion. The security system of their host country (USA) allows this type of capitalism. The USAians, who seem to not question such practices to encourage growth without much social good, are responsible for this dystopian statuses existance and continue to degrade human existence not only in their country but in most other countries. The exception are those countries who have put in safeguards and nurtured their own tech giants.

Any challenge to what these tech giants have usurped and now own is to take legal action which except for a few is beyond[23] the economic means and personal energy, commitment and moral resources of the rest of us.

4 WAITING FOR A SOLUTION

The protection of privacy, a human right, under threat from tech giants and goblins that they create requires some action. This could be either in the form of political and legislative and the form would be regulations and legislation with sizable penalties proportional to the income of the culprit, taxing the income etc. Another approach to be used is to set up national service for what now has become a way of many communications. The third approach, presented in the next section, is a technical solution to render the tech giants obsolete!

Some opinion expressed in the press for handling the tech giants is to recognize the service they provide as public service and either provide a national service under the control of an independent neutral organization and/or socialize them[82]. They have monopolized a number of services that they have usurped or re-engineered and made the population addicted to them. The addiction is evident in the homes, offices, public places and social get togethers where everyone constantly glances at their hand held devices[68]. These addicts are waiting for the next shot! No one seems to have recognized this addiction.

Waiting for a political solution is like En attendant Godot [14] but Godot never comes. The bent politicians are not in a hurry nor seem to have the moral strength to breakup these tech giants. The addiction that has been created with the so called free services has kept the politicians at bay. No thought has been given in any government to set up a national email service as an essential public infrastructure much as health, postal, road, school or train service. Even the tel-comm service is regulated in most countries. Since the internet depends on the tel-comm service it should be regulated with the tech giants at least held responsible for the contents. They should be taxed on their earnings in the jurisdiction where it is earned; there should be a penalty for the jobs that are shipped outside the country and for importing and exporting data. The tax should be at a progressive rate where the majority of the excess profit is taxed. This may encourage the tech giants to set up jurisdictional data farms to serve local emails, social contents.

Douthat[30] compares the western internet dominated by the USAian tech giants and the Chinese one dominated by the central government. The result in the western is the addiction generated by the internet and the control of it by a few corporations which at times work with the government and mistakes made on it are magnified. Lies and fake news are spread by it and real news is, by repetition from the top, labeled as fake news[30].

Cryptocurrency has evolved much later than search engines. Its spread is liable to upset the financial sector and the basis for the support of the political system everywhere much as the so called open internet has done by concentrating the imperialistic nature in the hands of a few tech giants all under the USAian form of capitalistic protection. However, the move to regulate Cryptocurrency has already begun in the form of legislators in various parts of the world. Regulation of the tech giant to respect the privacy of its users and not exploit their personal information to manipulate them is missing.

4.1 A possible start

One of the principals of privacy in the European Union's General Data Protection Regulation(GDPR) is that a person is the owner of her data and she has the right to decide who can use it and how. Regardless of where and how the data is shared, it can be amended, deleted or she could determine who and how it would be accessed [32]. GDPR went into effect in the EU in May 2018[42]. Its objective is to give control to individuals over their personal data and requires any organization who collects and controls personal information to have in place appropriate measures, both technical and logistic, to implement the data protection principles.

Such organizations are required to disclose their legal basis and purpose of data collection operations and have publicized the period of data retention and and the sharing of it with third parities. The data collecting organization are required to provide, to any data subject on request, a portable copy of the data collected in a common format. The data subject has the right to have their data corrected or even deleted. There are penalties for violation of this regulation, Under the violation of this regulation, recently France has fined Google 5.7 Million USD[75].

In the few months of coming into force of GDPRGDPR, the USAian government is finally waking up to some form of legislation for consumer privacy[59], driven ironically, not due to concern for consumer privacy but as a another component of high tech competition as outlined by Apples CEO[21], [50]. In the meantime activists are filing an increasing number of complaints under the GDPR[6]. In spite of the protection afforded by GDPR, it still allows the fundamental rights of the data subject not to override the business' legitimate interest of the data processor!

GDPR applies only to the EU, but given the scale of the market, many companies are deciding it's easier – not to mention a public relations win – to apply its terms globally. The problem is that even if there is a directive, even from a court, tech giants seem to consider themselves immune to these. A very recent example of this concerns a ban put in by a New Zealand court to name an accused killer. The local media companies, against who the court could take action, use resources to make sure such court bans are respected not only by themselves but also by their own social media channels. Google which does not apply bans globally and in line with this policy of geo-blocking (which is basically not being bound by local blackouts globally but only in the jurisdiction concerned) had emailed it out to users, apparently not in New Zealand, who had signed up for “what's trending in New Zealand”[57].

The effect of GDPR is being felt on this side of the Atlantic and accessing, for example a proper notice about cookies and use of analytics has to be given to EU citizens when they access USAian web sites: as usual there is a agree or not agree option! As usual it is too tempting to agree instead of looking at the privacy policy, third party partners or terms of service which are many pages long as pointed out earlier.

Competition by having another tech start-up to provide similar service seems hardly possible. Because of the large share of the market another similar OSN, even one such as Google+ did not succeed. Other avenues being used in the EU is to allow competition by blocking the tech giants from buying start-ups who may become a serious challenger some day. Such acquisitions have been allowed to proceed in the USA to date: buying of WhatAp and Instagram by Facebook are examples. The European model where the dominant giants are forced to share the data [32] goes back to the conclusion of the Workshop A held in April 1995 which recommended search engines share information[9].

5 WAY OUT OF THE PRIVACY AND SECURITY TRAP

The curernt situation where a small number of Usaian tech giants are controlling the web, all human knowledge and experience; they are manipulating awareness and beliefs to serve their aim of continued domination and maximizing profits. Their huge profits allow them to buy out any potential competition and are moving in new directions every day. This, after all has important elements in common with imperialism and totalitarianism. No surprise then that a country which experienced the latter most dramatically, Germany, has some of the strongest laws to safeguard privacy. Even still these efforts are merely corrective and merely polices the problem; not solve it. To actually overcome this system, a new solution is needed.

It is unlikely that many politicians who are heading governments or are part of the government have much motivation to do anything about privacy. The existing laws have no teeth and the tech giants are happy to put up the three big Ds(deny, deflect, delay). Each year they can delay the action, they are more established, made a few more billions and were able to finance more elections and place their men(mostly) in the drivers seat.

There are many political ideas put forward by various aspiring politicians in the western world. This is so in the prelude to the USAian 2020 presidential election. They include breaking up the tech giants, giving more control to the users of their data, making the algorithms transparent etc. None of this may work; take for instance making the algorithms more transparent; most users who don't even read the privacy agreement would not be able to understand the working of the algorithms. It is also doubtful that the tech giants would ever be willing to make their algorithms transparent.

The other idea is to increase competition; however this is also a no starter. The tech giants have big market capitalization and have politicians in their pockets. They make all possible effort to influence politicians since they have direct lines to the ministers and presidents. As a result we are proposing here a method to turn the clock back and bring home all communications and the data that is shared.

5.1 Lifting the cloud

Most users of the ‘free' services would not have read the privacy or the data-use policy when they sign up for these services. Reading these policies which are many pages long would be confusing with all their exceptions, and fighting any of its effect leads to years of battle in courts as is evidenced by the case cited earlier; such drawn out cases would exhaust the emotional energy of most users.

Web is a relatively recent way of doing things and as in many facets of human existence the way to do things swings from one way to another like a pendulum. Computing is no exception. We started with the idea of a ‘one of computing system' which would have been used to produce useful mathematical tables which would be printed and shared. In reality, this is not what happened. Computers were developed as a proof of concept and from there went to become what was called “main frames' - expensive and bulky systems. They were time shared by many users locally or remotely using dedicated telecommunication lines.

In the nineteen-sixties there were two trains of developments. A family of main frames were affordable enough to be used by many organizations to have their own computer systems and software development teams. At the same time mini-computers were developed to be used by smaller organizations and labs. The mini-computers evolved into the mini-computers and personal computer (PC) in the late 1970s and many people were able to have a personal desktop to do their own processing. The personal data was housed in the hard disk of the PC. Development of the hard disk technology allowed increase in speed and capacity. It was possible to store all local information locally.

The development and the spread of internet in the 1980s and then the world wide web starting in the early 1990s along with the graphical browser allowed the non-tech savvy person to be connected. The misdirection of the web by mainly commercial interests and the opportunity to claim uncharted territories prompted many tech buccaneers and geeks of the "dot.com" craze to start violating unwritten traditions and using and introducing surveillance tools, and thereby were able to amass huge troves of information.

The lack of the postal services to see electronic mail as a new public postal service, the ignorance and self-interest of politicians allowed the lack of regulation in the new domain allowing ownership of personal information of hundreds of million of individuals. The first incursion of private venture capitalists were in the domain of web search and email and the early companies included Altavista, Yahoo. Excite. Lycos. Even though web search engines started appearing in 1993, it was a later entry which captured the search market. Even though most search engines produce similar results, the habits and default setting in browsers tend to prioritize one.

As pointed out in [99] the concentration of data by such organizations is making it difficult for competition to be effective. The EU has ruled against Google many times in recent years; all of these are fought in the courts and the monopoly continues. The habits of people to flock to a system where others are and hence believe to be a better system has worked against titans as Google was forced to shutdown Google+ their social network. Not waiting for the breakup these tech giants and believing that less is better we propose here for users to take back control of their data, lives and privacy by offering them to host their own email and web server and setting up their own social network.

In a previous work we have pointed out the privacy issues with the increasing number of IoTs which transmit personal information to the servers of the makers of the IoTs. The key there was Heimdaller and the setting up of a Software Assurance Agency(SAA) [1]. This agency, is an independent one and requires that any device manufacturer must submit all software and updates to it for verification. It is independent and hence not run by a tech giant. No software unless it is certified for suitability would be certified. Unlike the 'stores' run by tech giants, SAA does not get a percent of the revenue for the software; however it charges a fee based on the size of the corporation. It is felt that there is a need for an independent organization such as SAA for the software industry much like the certification authorities CSA and UL. Here we propose to extend Heimdaller to not only monitor the IoTs but also act as a server for a personal email system and the web.

There are many systems that allow users to create their own web pages: an example is Facebook! Considering the number of articles, and litigations it has generated it is time that instead of giving away all this information to a corporation and sharing it with strangers, a personal web server could be used to allow the personal web page accessible only to the immediate family and friends.

The fact that a micro processor such as Raspberry is very affordable and is suitable for driving a personal email and web server with very little load and bandwidth need; that solid state memory and drives are now very affordable and could provide sufficient secure storage for the family server. The system would have its own storage and backup system; hence all storage of the family data, emails, web pages, comments etc. would be stored locally and there would be no need to use a cloud and thus deprive tech giants of the free raw material(data) and an opportunity to mine this information for their own profit. The proposed system, hence, include processing and storage. With a cheap processor such as Raspberry 2 and SSD the modem functionality required in private homes to connect to the internet through the intermediary of an ISP takes on the function not only of a sentry but also of a data vault.

As Arendt [5] in her chapter on Imperialism, talking about Cecil Rhodes, quoting the words of Millen "expansion is everything"[67]. Rhodes, looking at the stars and planet fell into despair since he wanted to annex all the planets for the British empire that he adored. Much like Rhodes some of the tech giants consider growth to be the good thing regardless of the collateral harm it does[66]. For instance Facebook knew that its platform could expose someone to bullying and coordinates terrorist attacks, This blunt memo by Bosworth recognizes that and noted that "The best products don't win. The ones everyone use win."[66]. With products like his, more users are attracted and they invite yet more!

5.2 Proposal for a Technical Solution

Breaking up the tech giants is not going to happen soon nor would an alternate commercial service start with the monetary power of the existing tech giants. They have the resources and staying power to bankrupt, buy and squash competition[98]. They have hundreds of lawyers working for them and connections to the highest level of the governments. The proposed system includes a modest processor, an email and web server and a light weight database and a new generation of modem router. The system addresses the biggest source of privacy violation: email and web presence.

Our proposal is simple; add the functionality of an email server and a web server to the modem-wireless router that mostpeople now have in theirhome of small office. This requires the adding of simple interfaces to allow even the most non-tech savvy user to mange these servers. All emails will originate in the users' owned system; the personal web server would host the persons web pages and all the contents would be stored locally. Access to the web server would be limited and the data could be shared as appropriate with various level of security. Only invited persons would be able to access any contents and since the user is in control of the web server and all its contents, she has the full control. Heimdaller is the gate keeper and all interaction of the Internet and IoTs including those coming from the users and the IoT maker goes through the gatekeeper. All software updates, have to be submitted to the SAA which verifies them and if they pass the tests of functionalities, it is

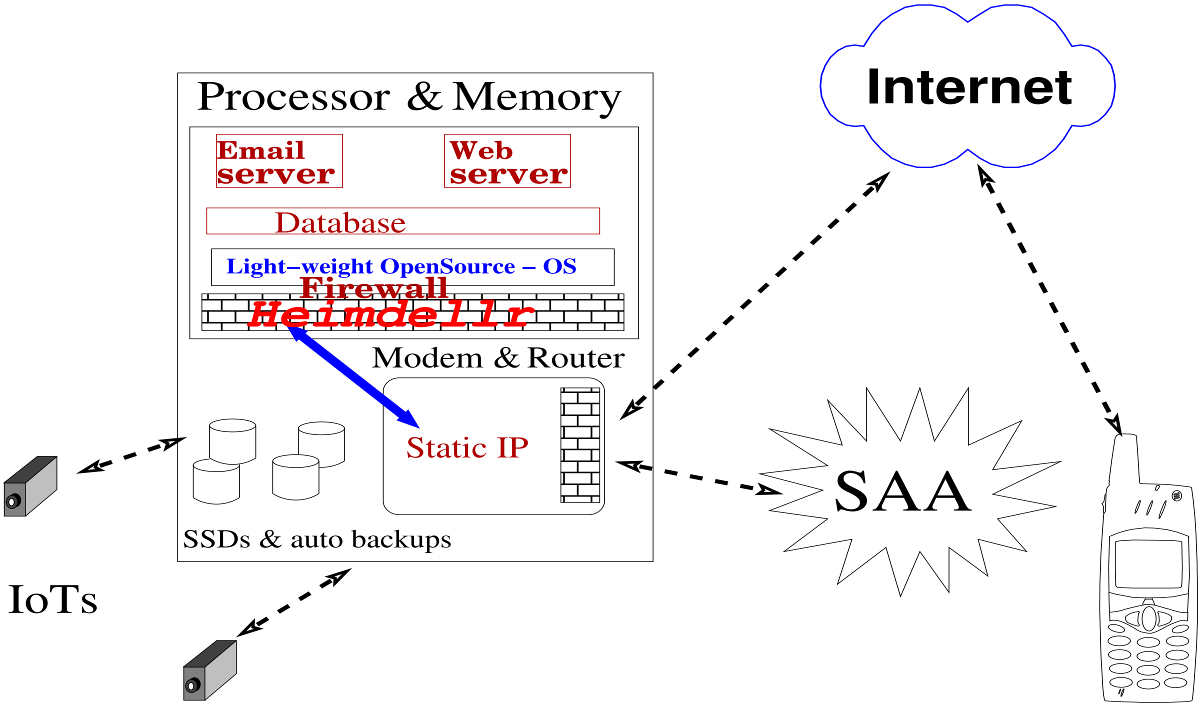


Figure 1: Proposal for a Technical Solution

certified and accessible to Heimdaller. Only SAA verified software could be installed in the the system

There would no longer be the need of any tech giants to provide email or web service. Technology has progressed to such an extent that these services could be incorporated in a device many home owners already have and its cost would be no more than that of a latest mobile device. The system would provide, each family in their own home an email server, a web server with their own family pages where they can post their news and share it with family and friends. The web server and the user interface would be such that expertise in making web pages would not be required. It is expected that the basics of internet usage, emailing, on-line chatting etc. would become programs in school. By including encryption in both the email and web contents, the leak of contents by eavesdropper is avoided.

The above development would mean that the not-really free services offered by the tech mammoths would not be required. So instead of waiting for a political solution from bent politicians, we are proposing a technical solution which would be created and maintained by the volunteer open source community and financed by required contributions from corporations making devices or software and donations by users.

6 CONCLUSION

The current practice of tech giants can only be neutralized by a technical solution where, their service would not be required. Once each family and businesses have their own static IP address and a hardnedconnection to the interenet with a server that provides emailand web service one becomes independent.The web server would not allow any robots and the gatekeepr, Heidammlalr would not allow any untrusted/uncertified software to be installed in the system. The development of such a sytem is the next challenge of the academic community!

Acknowledgement

The author is indebted to scores of commentators and op-ed writers who have been able to see the privacy being attacked by the tech giants. He also likea to acknowledge the many intresting discussion he had with Drew Desai (Univ. of Ottawa) and his valuable suggestions for the final format of this paper.

Appendix: The web from an early participant

Soon after the so called official inauguration of the world wide web(WWW) in the form of the first WWW meeting in Geneva, a flurry of activities were held in U. S. A. This included a rushed announcement by the National Center for Super-computing Applications(NCSA) of ‘Mosaic and the Web’ conference, which was renamed WWW II, and was held in Chicago. Whereas the first was announced by Robert Cailliau the second was spearheaded by NCSA Mosaic[78]. One of the early resolutions of these two meetings was raised in the Navigational and Priority workshops held during the first world wide web meeting (WWW-1) in Geneva in 1994. Other activity in the first days of the web was one in July 1995: it being a forum held by the USA National Science and Technology Council’s Committee on Information and Communication in Lister Hill Center (Bethesda, MD) entitled America in the Age Of information. A number of White Papers were presented[8]; looking through the list one finds that none of the white papers had touched on the issue of privacy. There was, but one, presentation on security.

During the subsequent early WWW meetings, some of the people involved in the navigation priority workshop devised various mechanisms for search in the new web. This included the WebJouornal [10] the support of robots and soon thereafter the early search engines. During WWWIII, in Darmstradt, the pioneers of the early search engines felt that to provide for the financial needs of the search engines, the side panel to display paid publicity would be appropriate. This way the paid publicity would be separated from the search results. This was the method used until a late arrived: initially, this new system was idealistic but soon became, under pressure from the venture capitalists, one of the leaders of the what has been termed the digital gangsters. All these systems are based on collecting huge amounts of personal information about the users, be it from free emails, or postings made on one of the online social networks (OSN)

REFERENCES

[1] Aksoy, Ayberk, Desai, Bipin C., 2019. Heimdallr: A system design for the next generation of IoTs In Proceedings of International Conference on Industrial Control Network and System Engineering Research (ICNSER2019) ACM, New York, NY, USA, 10 pages https://doi.org/0.1145/3333581.3333590

[2] ACM, CE2016: Computer Engineering Curricula 2016, https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf

[3] ACM, CS2013: Curriculum Guidelines for Undergraduate Programs in Computer Science, https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf

[4] ACM, Curricula Recommendation. https://www.acm.org/education/curricula-recommendations

[5] Arednt, Hannah, The Origin of Totalitarianism, Meridian Book, 1951, p 124

[6] BBC, Amazon, Apple and Google face data complaints, https://www.bbc.com/news/technology-46944694 last accessed May 6, 2019

[7] BBC, Google hit with €4.3bn Android fine from EU, Jul. 18 2018 https://www.bbc.com/news/technology-44858238 last accessed May 6, 2019

[8] AAI, America in the Age of Information, July 6-7, 1995, Lister Hill Center, Bethesda MD, http://users.encs.concordia.ca/ bcdesai/Age-of-Information-July-1995.pdf

[9] Desai, Bipin C., Pinkerton, Brian, Workshop A: Web−wide Indexing/Semantic Header or Cover Page, Summary, April 10, 1995, http://users.encs.concordia.ca/ bcdesai/web-publ/www3-wrkA/www3-wrkA-proc.pdf, also available from Spectrum Repository: https://spectrum.library.concordia.ca/985374/1/WWW-III-WrkShpA.pdf last accessed May 6, 2019

[10] Desai, Bipin C., Swiercz, Stan, WebJournal: Visualization of a Web Journey, June 1995, http://users.encs.concordia.ca/˜bcdesai/web-publ/WebJournal.pdf, Last accessed, May 7, 2019

[11] Desai, Bipin C., IoT: Imminent ownership Threat. In Proc. of 21st International Database Application & Enginnerring Symosium, Bristol, UK, July 2017 (IDEAS 2017), 8 pages. https://doi.org/10.475/3105831.3105843

[12] BELL CANADA AND LYCOS ANNOUNCE JOINT VENTURE, Feb 2, 2000, http://www.bce.ca/news-and-media/releases/show/bell-canada-and-lycos-announce-joint-venture

[13] Bell, Emily Mark Zuckerberg's Facebook mission statements hide his real aim, The Guardian, Mar.10, 2019, https://www.theguardian.com/media/commentisfree/2019/mar/10/markzuckerberg-facebook-mission- statements-hides-his-real-aim

[14] Beckett, Samuel En attendant Godott, Les Èditions de Minuit, Paris, 1952

[15] CBC Radio Facebook has become one of world's 'most dangerous monopolies, https://www.cbc.ca/radio/thecurrent/the-current-for-may-10-2019-1.5129874/friday-may-10-2019-full-transcript-1.5131529

[16] Cadwalladr, Carole, My TED talk: how I took on the tech titans in their lair, Guardain, Apr. 21, 2019, https://www.theguardian.com/uk-news/2019/apr/21/carole-cadwalladr-ted-tech-google-facebook-zuckerberg-silicon-valle

[17] Cadwalladr, Carole, Campbell, Duncan Revealed: Facebook's global lobbying against data privacy laws, The Guardain, Mar. 2. 2019, https://www.theguardian.com/technology/2019/mar/02/

[18] Chisick, Chris Supreme Court of Canada Upholds BC Decision to Grant Worldwide De-Indexing Order Against Google, June 28, 2017 https://www.casselsbrock.com/CBNewsletter/Supreme_Court_of_Canada_Upholds_BC_Decision_to_Grant _Worldwide_De_Indexing_Order_Against_Google, https://www.canlii.org/en/ca/scc/doc/2017/2017scc34/2017scc34.html

[19] CIHI, Privacy and Security Risk Management Framework , h ttps://www.cihi.ca/en/about-cihi/privacy-and-security

[20] Confessore, Nicholas, Rosenberg, Matthew, Damage Control at Facebook: 6 Takeaways From The Times'sInvestigation, New York Times, Nov. 14, 2018, https://www.nytimes.com/2018/11/14/technology/ facebook-crisis-mark-zuckerberg-sheryl-sandberg.htm

[21] Cook, Tim, You Deserve Privacy Online. Here's How You Could Actually Get It, http://time.com/collection/davos-2019/5502591/tim-cook-data-privacy/

[22] Duffy, Andrew Trudeau and Liberals win majority in historic return to power Ottawa Citizen, Oct. 20, 2015 https://ottawacitizen.com/news/politics/justin-trudeau-and-liberals-stage-historic-return-to-power

[23] CBC, B.C. court approves class-action lawsuit against Facebook, May 30, 2014 https://www.cbc.ca/news/canada/british-columbia/facebook-class-action-lawsuit-launched-by-vancouver-woman-1.266046, Last accessed, May 6, 2019

[24] CBC, Facebook wins appeal to stop B.C. class-action lawsuit over privacy, Jun 19, 2015, https://www.cbc.ca/news/canada/british-columbia/facebook-wins-appeal-to-stop-b-c-class-action-lawsuit-over-privacy-1.3120849, Last accessed, May 6, 2019

[25] CTV, Supreme Court clears way to B.C. class-action against Facebook, June 23, 2017, https://www.ctvnews.ca/canada/supreme-court-clears-way-to-b-c-class-action-against-facebook-1.3473026, Last accessed, May 6, 2019

[26] Denham, Elizabeth, Report of Findings into the Complaint Filed by the Canadian Internet Policy and Public Interest Clinic (CIPPIC) against Facebook Inc. Under the Personal Information Protection and Electronic Documents Act, PIPEDA Report of Findings #2009-008, July 16, 2009, https://www.priv.gc.ca/en/opc-actions-and-decisions/investigations/investigations-into-businesses/2009/pipeda-2009-008/ Last accessed, April 18, 2019

[27] Dance, Gabriel J.X., et.al, As Facebook Raised a Privacy Wall, It Carved an Opening for Tech Giants, NY Times, Dec. 18, 2018, https://www.nytimes.com/2018/12/18/technology/facebook-privacy.html

[28] Gates, Dominic Flawed analysis, failed oversight: How Boeing, FAA certified the suspect 737 MAX flight control system Seattle Times, Mar. 17, 2019 https://www.seattletimes.com/business/boeing-aerospace/failed-certification- faa-missed-safety-issues-in-the-737-max-system-implicated-in-the-lion-air-crash/ Last accessed, May 19, 2019

[29] D'Onfro, Jillian, Google did not disclose security bug because it feared regulation, says report, CNBC, Oct 8 2018, https://www.cnbc.com/2018/10/08/google-reportedly-exposed-private-data-of-at-least-hundreds-of-thousands-of-plus-users.html

[30] Douthat Ross, The Only Answer Is Less Internet, New York Times, April 13, 2019. https://www.nytimes.com/2019/04/13/opinion/china-internet-privacy.html, Last accessed April 23, 2019

[31] Dent, Steve Report: Boeing's crucial 737 Max safety analysis was flawed EnGadget, Mar. 18, 2019 https://www.engadget.com/2019/03/18/boeing-737-max-faa-certification-flaws/ Last accessed April 23, 2019

[32] The Economist, The future of big tech Why big tech should fear Europe, The Economist, Mar 23rd 2019, https://www.economist.com/leaders/2019/03/23/why-big-tech-should-fear-europe

[33] Data Policy - Facebook, https://www.facebook.com/ about/privacy/update

[34] Terms of Service – Facebook, https://www.facebook.com/legal/terms/update

[35] Fleshman, Glenn, FTC Considers Record Fine for Facebook Over Violation of User Privacy Agreement It Made in 2012, Report Says, http://fortune.com/2019/01/18/facebook-privacy-ftc-record-fine-considered/

[36] Frenkel, Sheera, et. al. Delay, Deny and Deflect: How Facebook's Leaders Fought Through Crisis NY Times, Nov. 14, 2018, https://www.nytimes.com/2018/11/14/technology/facebook-data-russia-election-racism.html

[37] COPPA, Children's Online Privacy Protection Act, https://www.ftc.gov/enforcement/statutes/childrens-online-privacy-protection-act

[38] FTC, Privacy and Security, https://www.ftc.gov/tips-advice/business-center/privacy-and-security

[39] FTC, Careful Connections: Building Security in the Internet of Things, https://www.ftc.gov/tips-advice/business-center/guidance/careful-connections-building-security-internet-things

[40] Goodwin, Bill , et al., Facebook asked George Osborne to influence EU data protection law, Computer Weekly, Mar. 2, 2019, https://www.computerweekly.com/news/252458229/Facebook-asked-George-Osborne-to-influence-EU-data-protection-law

[41] GDPR, What is GDPR and how will it affect you?, https://www.theguardian.com/technology/2018/may/21/what-is-gdpr-and-how-will-it-affect-you

[42] GDPR, General Data Protection Regulation https://gdpr-info.eu/

[43] Graham-Harrison, Emma, Inside Facebook's war room: the battle to protect EU elections, Guardian, May 5, 2019, https://www.theguardian.com/technology/2019/may/05/facebook-admits-huge-scale-of-fake-news-and-election-interference

[44] Guruswamy, Menaka India's Supreme Court Expands Freedom NY Times, Sept. 10, 2017 https://www.nytimes.com/2017/09/10/opinion/indias-supreme-court-expands-freedom.html?

[45] Guliani, Neema Singh, W Should Be Able to Take Facebook to Court, NY Times, Jan.. 6, 2019, https://www.nytimes.com/2019/01/06/opinion/facebook-privacy-violation.html

[46] GPS, Global Positioning System History, https://www.nasa.gov/directorates/heo/scan/ communications/policy/GPS_History.html Last accessed Apr. 7, 2019

[47] Facebook uploaded email contacts of 1.5m users without consent, The Guardian, Apr. 18, 2019, https://www.theguardian.com/technology/2019/apr/18/facebook-uploaded-email-contacts-of-15m-users-without-consent

[48] Hughes, Chris It's Time to Break Up Facebook, NY Times. May 9, 2019, https://www.nytimes.com/2019/05/09/opinion/sunday/chris-hughes-facebook-zuckerberg.html

[49] Hart, David, On the Origins of Google:, August 17, 2004, https://www.nsf.gov/discoveries/disc_summ.jsp?cntn_id=100660

[50] Hem, Alex, Apple chief calls for laws to tackle 'shadow economy' of data firms, Jan. 17, 2019, https://www.theguardian.com/technology/2019/jan/17/apple-chief-tim-cook-calls-for-laws-to-tackle-shadow-economy-of-data-firms

[51] Huish, Robert, Balazo Patric, Unliked: How Facebook is playing a part in the Rohingya genocide, The Conversation, Jan. 2, 2018, https://theconversation.com/unliked-how-facebook-is-playing-a-part-in-the-rohingya-genocide-89523

[52] Insley, Jill FarmVille user runs up £900 debt The Guardian, Apr. 7, 2010 https://www.theguardian.com/money/2010/apr/07/farmville-user-debt-facebook

[53] Johnson, Bobby, Privacy no longer a social norm, says Facebook founder , The Guardian, Jan 11, 2010, https://www.theguardian.com/technology/2010/jan/11/facebook-privacy

[54] Jacobsson, Andreas; Davidsson, Paul, Towards a Model of Privacy and Security for Smart Homes. Proc.: 2015 IEEE 2nd World Forum on Internet of Things (WF-loT), p. 727-732 URL: https://doi.org/10.1109/WF-IoT.2015.7389144

[55] Jenkins, Simon, Facebook is out of control and politicians have no idea what to do The Guardian, Feb, 18, 2019, https://www.theguardian.com/commentisfree/2019/feb/18/ facebook-powerful-politicians-commons-abuse

[56] Lycos, https://en.wikipedia.org/wiki/Lycos

[57] Manhire, Toby, New Zealand courst banned naming Grace Millane'e accused killer,. Google just emailed it out 13 Dec. 2018, https://www.theguardian.com/world/2018/dec/13/new-zealand-courts-banned-naming-grace-millanes-accused-killer-google-just-emailed-it-out

[58] Madrigal, Alexis C. , What We Know About Facebook's Latest Data Scandal, The Atlantic, June 4, 2018, https://www.theatlantic.com/technology/archive/2018/06/what-we-know-about-facebooks-latest-data-scandal/561992/

[59] Meyer, David, In the Wake of GDPR, Will the U.S. Embrace Data Privacy?, Fortune, Nov. 29. 2018, http://fortune.com/2018/11/29/federal-data-privacy-law/

[60] Morozov, Evgeny, It's not enough to break up Big Tech. We need to imagine a better alternative, Guardian, May 11, 2019 https://www.theguardian.com/commentisfree/2019/may/11/big-tech-progressive-vision-silicon-valley

[61] Mullin, Joe Privacy lawsuit over Gmail will move forward Aug. 16, 2016 https://arstechnica.com/tech-policy/2016/08/privacy-lawsuit-over-gmail-will-move-forward/

[62] King, Mark Parents told to beware children running up huge bills on iPad and iPhone game apps Guardian, Jan. 12, 2013 https://www.theguardian.com/technology/2013/jan/12/parents-children-in-app-purchases

[63] McLaren Leah Is Elizabeth Denham the Only Person Powerful Enough to Take on Facebook? The Walrus, Apr. 18, 2019 https://thewalrus.ca/is-elizabeth-denham-the-only-person-powerful-enough-to-take-on-facebook/

[64] Martin, Nicole Was The Facebook '10 Year Challenge' A Way To Mine Data For Facial Recognition AI?, Forbes, Jan. 17, 2019. https://www.forbes.com/sites/nicolemartin1/2019/01/17/was-the-facebook-10-year-challenge-a-way-to- mine-data-for-facial-recognition-ai#4b56c3fe5859

[65] The history of Mobile phones, https://en.wikipedia.org/wiki/History_of _mobile_phones, last accessed April, 2019

[66] McNamee, Roger I Mentored Mark Zuckerberg. I Loved Facebook. But I Can't Stay Silent About What's Happening, The Time, Jan ., 17, 2019, http://time.com/5505441/mark-zuckerberg-mentor-facebook-downfall/

[67] Millen, Sarah Gertrude, Rhodes, London, 1933, p. 138.

[68] Nancherla, Aparna, Lee, Christopher, The Infinite Scroll, New York Times, April 13, 2019, https://www.nytimes.com/2019/04/13/opinion/sunday/the-infinite-scroll.html,

[69] Privacy Commissioner launches Facebook investigation. March 20, 2018, https://www.priv.gc.ca/en/ opc-news/news-and-announcements/2018/nr-c_180320/

[70] OSN, Why the UK is taking on social networks over child safety, https://www.theguardian.com/technology/2019/feb/06/why-uk-is-taking-on-social-networks-child-safety, Last accessed April 16, 2019

[71] Le NPD réclame à la commissaire au lobbying une enquête sur Facebook, La Presse, Mar. 4, 2019, https://www.lapresse.ca/actualites/politique/politique-canadienne/201903/04/01-5216953-le-npd-reclame-a-la-commissaire-au-lobbying-une-enquete-sur-facebook.php

[72] Pegg, David, Facebook labelled 'digital gangsters' by report on fake news, Guardian, Feb. 18, 2019, https://www.theguardian.com/technology/2019/feb/18/facebook-fake-news-investigation-report-regulation-privacy-law-dcms

[73] Joint investigation of Facebook, Inc. by the Privacy Commissioner of Canada and the Information and Privacy Commissioner for British Columbia PIPEDA Report of Findings #2019-002, April 25, 2019 https://www.priv.gc.ca/en/opc-actions-and-decisions/investigations/investigations-into-businesses/2019/pipeda-2019-002/ Last accessed April 30, 2019

[74] Paul, Kari, Facebook security lapse affects millions more Instagram users than first stated. Guardian, April 18, 2019, https://www.theguardian.com/technology/2019/apr/18/instagram -facebook-password-lapse-privacy-breach-data-exposed- Last accessed, April 18, 2019

[75] Price, Emily, France Fines Google $57 Million For GDPR Violation, Jan. 21 2019, http://fortune.com/2019/01/21/france-fines-google-57-million-for-gdpr-violations/

[76] Privacy Shield, Key New Requirements for Participating Companies, Informing individuals about data processing, https://www.privacyshield.gov/Key-New-Requirements

[77] Privacy Shield, https://www.privacyshield.gov/Program-Overview

[78] WWW1 First International Conference on the World-Wide Web, https://en.wikipedia.org/wiki/First_International _Conference_on_the_World-Wide_Web

[79] Rushe, Dominic Google: don't expect privacy when sending to Gmail Teh Guardian, Aug 15, 2013 https://www.theguardian.com/technology/2013/aug/14/google-gmail-users-privacy-email-lawsuit

[80] Ressa Maria, Facebook Let My Government Target Me. Here's Why I Still Work With Them, The Time, Jan. 17, 2019, http://time.com/5505458/facebook-maria-ressa-philippines/

[81] Ryan, Marc, Warzel, Charlie, Kantrowitz. Alex, Growth At Any Cost: Top Facebook Executive Defended Data Collection In 2016 Memo — And Warned That Facebook Could Get People Killed, Buzzfeed, March 29, 2018, https://www.buzzfeednews.com/article/ryanmac/growth-at-any-cost-top-facebook-executive-defended-data, Last accessed April 12, 2019

[82] Srnicek, Nick. The only way to rein in big tech is to treat them as a public service, The Guardian, Apr 23, 2019, https://www.theguardian.com/commentisfree/2019/apr/23/big-tech-google-facebook-unions-public-ownership

[83] Stueck, Wendy, Former information commissioner Elizabeth Denham was one of first to raise concerns over Facebook data, Globe and Mail, March 25, 2018, https://www.theglobeandmail.com/canada/british-columbia/article-former-information-commissioner-elizabeth-denham-was-one-of-first-to

[84] Turow, Joseph, Let's Retire the Phrase 'Privacy Policy', New York Times, Aug. 20, 2018. https://www.nytimes.com/2018/08/20/opinion/20Turow.html

[85] Turow, Joseph, Google Still Doesn't Care About Your Privacy Fortune, June 28, 2017 http://fortune.com/2017/06/28/gmail-google-account-ads-privacy-concerns-home-settings-policy/

[86] Tiktok: India bans video sharing app, https://www.theguardian.com/world/2019/apr/17/ tiktok-india-bans-video-sharing-app, Last accessed April 16, 2019

[87] TikTok video-sharing app fined for collection of children's data , The Guardian, Feb. 28, 2019 https://www.theguardian.com/technology/2019/feb/28/ tiktok-video-sharing-app-fined-for-collection-of-childrens-data, Last accessed April 16, 2019

[88] Tufekci, Zeynep, Zuckerberg's So-Called Shift Toward Privacy, NY Times, Mar. 7, 2019, https://www.nytimes.com/2019/03/07/opinion/ zuckerberg-privacy-facebook.html

[89] Disinformation and 'fake news': Final Report, Digital, Culture, Media and Sport select committee, Feb, 2019, https://publications.parliament.uk/pa/cm201719/cmselect/ cmcumeds/1791/179102.htm

[90] von Scheel, Elise, Facebook pressured Canada to ease up on data rules, U.K. reports say, CBC News, Mar 03, 2019, https://www.cbc.ca/news/politics/facebook-canada-data-pressure-1.5041063

[91] Valentino-de Vries, Jennifer, Tacking Phones, Google Is a Dragnet for the Police, NY Times, April 4, 2019, https://www.nytimes.com/interactive/2019/04/13/us/google-location-tracking-police.html

[92] Vaidhyanathan, Siva Facebook's new move isn't about privacy. It's about domination, Guardain, Mar. 7, 2019, https://www.theguardian.com/commentisfree/2019/mar/07/ facebook-privacy-domination

[93] Warzel, Charlie, Thompson, Stuart A. Tech Companies Say They Care, NT Times April 10, 2019, https://www.nytimes.com/interactive/2019/04/10/opinion/tech-companies-privacy.html, Last accessed April 17. 2019

[94] Waterson, Jim, Obscure pro-Brexit group spends tens of thousands on Facebook ads, Guardain, Jan. 14, 2019, https://www.theguardian.com/politics/2019/jan/14/obscure-pro-brexit-group-britains-future-spends-tens-of-thousands-on-facebook-ads

[95] Waters, Richard, Murphy, Hannah, Stacey, Kiran, Social Media's Reckoning?, Financial Times, April, 13-14, 2019, p6

[96] Wu, Tim, How Capitalism Betrayed Privacy, NY Times, April 10, 2019 https://www.nytimes.com/2019/04/10/opinion/sunday/privacy-capitalism.html

[97] Weinberg, Zoe A. Y. Google Settles Buzz Lawsuit The Harvard Crimson, Sep. 7, 2010 https://www.thecrimson.com/article/2010/9/7/google-mason-privacy-settlement/

[98] Yglesias Matthew The push to break up Big Tech, explained Vox, May 3, 2019 https://www.vox.com/recode/2019/5/3/18520703/big-tech-break- up-explained

[99] Zuboff, Shoshana, The Age of Surveillance Capitalism, Jan, 2019, pp179, ISBN 97801-61039-564-4

Supervised Learning Methods Application to Sentiment Analysis

Sergio Altares López
University of Alcala
Alcalá de Henares, Madrid, Spain
sergio.altares@edu.uah.es

Juan J. Cuadrado-Gallego
University of Alcala
Alcalá de Henares, Madrid, Spain
jjcg@uah.es

ABSTRACT

The field of artificial intelligence (AI) is constantly growing and finding new ways to solve real world problems. One of the AI knowledge and research fields is natural language processing (NLP) which attempts to categorise and process human language data in an effort to utilise machines to understand humans.

Among the most used applications of NLP is Sentiment Analysis. This is because, in addition to other reasons, Sentiment analysis is about understanding how humans are feeling related to an action or event, what could give to the companies with an online presence the power to understand the opinions of their customers online.

A commonly used weighting factor to measure the perform of sentiment analysis is the *tfidf*. In this study we compare several supervised learning methods using the tfidf values in order to identify the most accurate model to analyse sentiment.

After we observe which is the best classifier based on metrics and other parameters we will do a real application of sentiment analysis with twitter data.

CCS CONCEPTS

• **Computing methodologies** → **Artificial Intelligence, Machine Learning**; • **Information Systems** → *Data Mining*;

KEYWORDS

Data Science, NLP, Machine Learning, Supervised Learning, Artificial Neural Networks, Sentiment Analysis.

ACM Reference Format:
Sergio Altares López and Juan J. Cuadrado-Gallego. 2019. Supervised Learning Methods Application to Sentiment Analysis. In *23rd International Database Engineering Applications Symposium (IDEAS'19), 10–12 June 2019, Athens, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3331076.3331086

1 INTRODUCTION

The field of artificial intelligence (AI) is constantly growing and finding new ways to solve real world problems [19]. One of the AI knowledge and research fields is natural language processing (NLP) which attempts to categorise and process human language data in an effort to utilise machines to understand humans.

Natural Language Processing is an artificial intelligence field focused on enabling computers to understand, process and act based on human languages, getting computers closer to a human level language understanding [14]. Some advances in Machine Learning have enabled computers to do many useful things using NLP techniques and deep learning [23] such as online language translators or semantic understanding [7].

One of the most popular and important uses of NLP is **Sentiment Analysis** [20]. With this technique we can build systems which attempt to identify and extract opinions or sentiments from oral speaking or written texts [2]. This type of analysis is very important for companies because they can take data from customer's opinions and thus make improvements to their businesses.

Since computers are not able to understand human expressions, we need to create a binary vector in which each word has its own position which is called **vectorize**.

Once it has been vectorized, a term frequency–inverse document frequency (tfidf) weighting factor can be applied [5]. This new vector based on word frequencies will be the input of all models we will train. It is calculated as the product of the term frequency and the inverse document frequency [13]. Since **m** is the word and **l** the document, **tf(m, l)** is the number of times that **m** appears in **l**. We can see the mathematical expression in **Equation 1**.

$$\mathrm{tf}(m,l) = \frac{\mathrm{f}(m,l)}{\max\{\mathrm{f}(m,l) : m \in l\}} \tag{1}$$

The term **idf** refers to the inverse frequency that consists of knowing if the word is common in a set of documents. This value, as we can observe in **Equation 2**, is obtained by dividing the total number of documents (**L**) by the number of documents that contains the word, and then the logarithm is taken [16].

$$\mathrm{idf}(m,L) = \log \frac{|L|}{|\{l \in L : m \in l\}|} \tag{2}$$

The final value is calculated as the product of both, as we can see in the **Equation 3** [22].

$$\mathrm{tfidf}(m,l,L) = \mathrm{tf}(m,l) \times \mathrm{idf}(m,L) \tag{3}$$

A high **tfidf** value signifies a high frequency of the word in the document and a small frequency of occurrence of the instance in the set of documents [9].

IDEAS'19, 10–12 June 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331086

In addition to NPL, in the broad world of the use of Artificial Intelligence, we find the application to data science. Data Science fundamentally consists of processing, analysing and creating models in order to extract information from data. Deeper within this field, machine learning is responsible for training computers to learn based on data. The types of machine learning methods could be divided in supervised, unsupervised and reinforcement.

Supervised learning requires that the machine be trained with previous data in order to obtain a model that can be applied to new input data. On the contrary, the unsupervised learning does not require training, it groups data in k groups to be able to make profiles or join data with similar behavior. Finally, reinforcement learning is based on rewards and penalties.

The present study aims to identify to most accurate supervised learning method for sentiment analysis. Performing this type of analysis is very useful for companies as it allows them to know consumers opinions and thus, they can use this information to improve products, departments or marketing strategies, among others. In this article we will analyze texts, taking into account different supervised learning methods, as well as an artificial neural network (ANN), in order to compare and identify the precision of each technique [3].

After this introduction the rest of the paper is structured in five sections. The next section we will define the data used in this experiment. The section number three is about the research methods used as well as metrics used to compare them. After that we will see the main results that we have seen during the experiment, we will discuss about these results and also we will test the best model with new twitter data. Finally we will see the conclusions that we can take out from this experiment.

2 DATASETS

The data used for the present study belongs to the corpus denominated *yelp labelled* and *amazon cells labelled* [15]. Both datasets have, as it can be seen in the **Table 1**, two columns called *Tweets* and *Labels*.

Table 1: Datasets [6]

Yelp Labelled	
Tweets	**Labels**
Wow... Loved this place.	1
Crust is not good.	0
The selection on the menu was great.	1

Amazon Cells Labelled	
Tweets	**Labels**
There is no way for me to plug it in here.	0
Good case, Excellent value.	1
What a waste of money and time!.	0

Since each corpus has 1000 lines of data correctly labelled, we will use a corpus to train and after that we will use the second corpus for validation. Validation with x_{test} will provide us with y_{pred}. Since we already have the correct labels in the validation set, we will compare y_{pred} and y_{test} in order to asses if the classification is correct.

Table 2: Twitter Real Application

Tweets
Have a great day as well! We have many more promos planned. Stay tuned!
I would be more than happy to further investigate this transaction.
That's AMAZING! Congrats on the 20th year. We greatly appreciate you as a member.

Once we have done the models training part with data from **Table 1**, we will perform a quick test using the best classifier with tweets which will be downloaded from the twitter API REST. In this case we will not have labels for each instance, as we can see in **Table 2**, that is what really happen on companies.

3 RESEARCH DISCUSSION

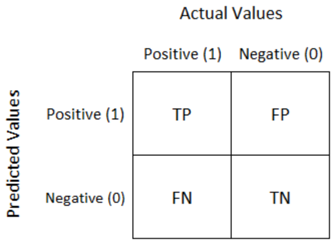
Supervised learning methods are algorithms which base their learning process on an accurately labeled training data set [4]. This means that for each occurrence of the training data set, we know the value of its target variable. Its use is limited to classification or regression [17]. In this article we will use these kind of methods to sentiment analysis so we can see which is the most accurately of them.

First of all, data will be vectorized as we saw in the previous section and a tfidf weighting factor is applied thus, the inputs of each classifier will be based on the frequency of each term.

After that, since we have two corpus correctly labeled, we will use one of them to perform the training process and the other one to test. The validation labels will be temporarily removed so once we have the predictions from the trained model we could compare them with the real labels, being able thus to extract data from correct predictions.

We will apply several supervised methods so we can see the differences between them, taking into account metrics as confusion matrix and Area Under Curve (AUC) [8]. The confusion matrix is a graphical visualization of how accurately is the model and is composed for four measures as we can see in **Figure 1**.

Figure 1: Confusion Matrix



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Being **TP** (True Positives) which means that the data was correctly predicted; **FP** (False Positives) indicates that some data were predicted as positive but it was a wrong prediction. **FN** (False Negative) is defined as data that are predicted as negative but are incorrect and **TN** (True Negative) that are data correctly predicted as negative.

Using the confusion matrix terms we can calculate other metrics that we will take into account to compare our models as *Recall* or *True Positive Rate* (TPR) which measures the rate of true positives. It is calculated as the following mathematical expression:

$$Recall = TPR = \frac{TP}{TP + FN} \tag{4}$$

Precision is another metric that measures performance related to positive and negative rates and is calculated as the following expression:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Normally precision and recall are combined in a metric called *F1* which can be calculated as the following equation:

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{6}$$

Another parameter we need to calculate AUC is the *False Positive Rate* (FPR) which we can calculate applying the next expression:

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

The other metric we are going to take into account is *Area Under Curve* (AUC) which measures the performance of the model. It is created by plotting the *True Positive Rate* (TPR) against the *False Positive Rate* (FPR) which is calculated as follows [12]:

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{8}$$

Once we calculate the metrics for each model and see which is the best one, we will download tweets from API REST to classify. In this case we will not have tagged each instance as companies work so we will use the best classifier based on metrics calculated before.

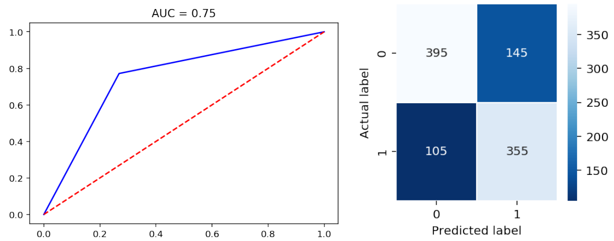
In the following subsections we will see all the supervised learning methods that we use for this experiment, seeing for each method the AUC and confusion matrix metrics.

3.1 Naïve Bayes

Naive Bayes is a probabilistic classifier based on the application of Bayes theorem and the independence hypothesis between the predictive variables. This classifier assumes that the presence of a particular feature is not related to the presence or absence of any other feature, it considers that every single feature contributes independently to the target probability [10].

Then, evaluating the model with confusion matrix metric, we can observe that in 750 cases the prediction was correct. These correct data are composed by 355 predicted correctly as positives and 395 predicted correctly as negatives. By the other hand, we find that in 250 cases the prediction was incorrect. These incorrect data are composed by 145 predicted incorrectly as positives and 105 cases which were predicted incorrectly as negatives.

Figure 2: Naïve Bayes Metrics



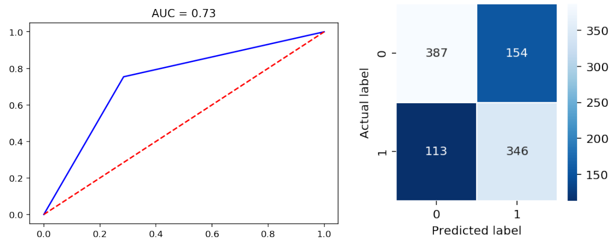
As we can see in the previous **Figure 2**, this classifier reaches an AUC of 0.75. It can be observed in the confusion matrix that there are more failures in the negative results than in the positive ones.

3.2 Support Vectorial Machine

Support Vectorial Machine (SVM) is a kind of supervised learning method which creates models that represent points in space, separating the classes into two spaces as wide as possible [1].

These separations form hyperplanes, defined as the vector between the two closest points of each class, which are called *support vectors* . When the new samples are in correspondence with the trained model, they can be classified to one of these classes. SVMs have the ability to construct a hyperplane or set of hyperplanes in a high dimensional space [10].

Figure 3: Support Vectorial Machine Metrics



Once we have trained our model, we can see that in 733 cases the prediction was correct. These correct data are composed by 346 predicted correctly as positives and 387 predicted correctly as negatives. By the other hand, we find that in 267 cases the prediction was incorrect. These incorrect data are composed in 154 predicted incorrectly as positives and 113 predicted incorrectly as negatives.

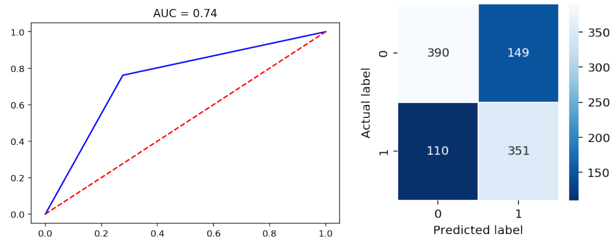
It can be seen in the **Figure 3** that the AUC that reaches the vector support machine algorithm is 0.73, classifying better the positive data than the negative.

3.3 Logistic Regression

Logistic Regression models have become an accepted method to obtain binary outcome variables [11]. It is a type of regression analysis used to predict the outcome of a categorical variable based on predictor variables. The main idea of Logistic Regression is that it approximates the probability of obtaining 1 or 0 with the explanatory variable x value.

After training the model we have that in 741 cases the prediction was correct. These correct data are composed by 351 predicted correctly as positives and 390 predicted correctly as negatives. By the other hand, we find that in 259 cases the prediction was incorrect. These incorrect data are composed by 149 predicted incorrectly as positives and 110 predicted incorrectly as negatives.

Figure 4: Logistic Regression Metrics

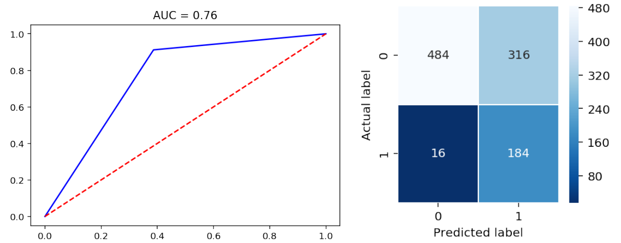


As we can observe in the **Figure 4**, the model reaches a 0.74 AUC and separates, with the same quality as the previous model, the positive and negative data.

3.4 Tree Decision

A Tree Decision is an artificial intelligence algorithm that is based on making diagrams. It consists on represent and categorize a series of conditions that occur successively [18].

Figure 5: Tree Decision Metrics



We can see that in 668 cases the prediction was correct. These correct data are composed by 184 predicted correctly as positives and 484 predicted correctly as negatives. By the other hand, we find that in 332 cases the prediction was incorrect. These incorrect data are composed by 316 predicted incorrectly as positives and 16 predicted incorrectly as negatives.

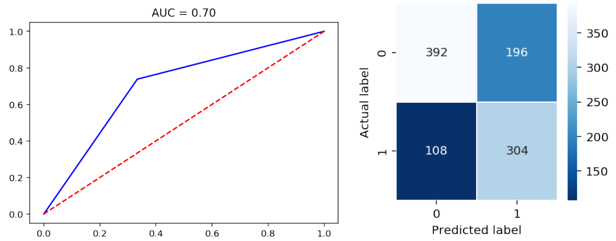
It can be seen that the AUC is 0.76. In contrast, we see that it classifies the majority of data as negative, so it has a great defect in terms of classification of positives, despite the high value of AUC.

3.5 Random Forest

A Random Forest algorithm consists of a combination of supervised predictive trees. Each tree independently relies on the values of a randomly tested vector and with that same distribution for each of them.

The model provides us that in 696 cases the prediction was correct. These correct data are composed by 304 predicted correctly as positives and 392 predicted correctly as negatives. By the other hand, we find that in 304 cases the prediction was incorrect. These incorrect data are composed by 196 predicted incorrectly as positives and 108 predicted incorrectly as negatives.

Figure 6: Random Forest Metrics

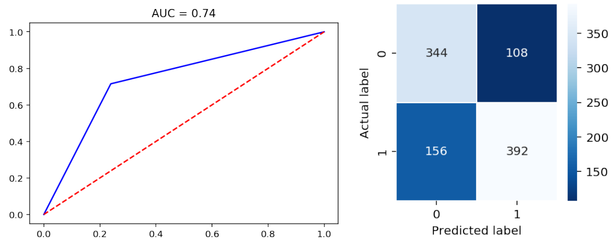


We see that the AUC is 0.70. This result is much lower than all of the previous algorithms which can also be seen in the confusion matrix.

3.6 Perceptron

Perceptron is an algorithm for supervised learning for binary classification. This classifier makes its predictions based on a linear predictor function, combining weights and the feature vector. It creates a hyperplane so if the training set Z is not linearly separable it will not separate correctly both classes.

Figure 7: Perceptron Metrics



After training the model we have that in 736 cases the prediction was correct. These correct data are composed by 392 predicted correctly as positives and 344 predicted correctly as negatives. By the other hand, we find that in 264 cases the prediction was incorrect. These incorrect data are composed by 108 predicted incorrectly as positives and 156 predicted incorrectly as negatives.

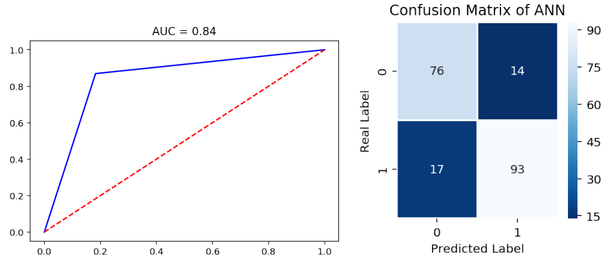
As we can see in the **Figure 4**, the model reaches a 0.74 AUC and separates, with the same quality as the previous model, the positive and negative data.

3.7 Multilayer Perceptron (MLP)

A Multilayer Perceptron (MLP) is an artificial neural network formed by multiple layers, in such a way that it has the capacity to solve problems that are not linearly separable. This provides a solution to the problem found in perceptron, as we saw in the previous section [21].

In this case, since the Neural Network training process is harder than other methods used before, we will train the MLP with both dataset, using for the validation process 10% of all data.

Figure 8: Multilayer Perceptron Metrics



The MLP has an input layer through which data enters on it, and three hidden layers connected to each other with a non-linear activation function in order to find non-linear patterns, in this case sigmoid. Finally we find the output layer which activation function is also sigmoid.

As we can see in **Figure 8**, the network reaches 0.84 of AUC. It can be seen in the confusion matrix that this algorithm has effectively classified both the positive and negative data.

4 RESULTS AND DISCUSSION

Reviewing only the AUC metric in **Table 3** after having tested several supervised learning methods, the highest result comes from the Tree Decision with an AUC of 0.76. However, with the lowest *precision rate* of 0.38 and 0.538 of *F1*, it is a classifier that has many failures in the positives.

Table 3: Metrics per Classifier

Classifier	F1	Recall	Precision	AUC
Naïve Bayes	0.739	0.771	0.71	0.75
SVM	0.721	0.753	0.69	0.73
Logistic Regression	0.730	0.761	0.70	0.74
Tree Decision	0.538	0.909	0.38	0.76
Random Forest	0.666	0.737	0.61	**0.70**
Perceptron	0.748	0.715	0.78	0.74
MLP	0.857	0.869	0.85	**0.84**

On the other hand, Naïve Bayes is a classifier with an AUC is 0.75. This result is a very small difference between the Tree Decision of 0.01, but it classifies in a more equitable way the negative and positive data. As we can see in **Table 4** Tree Decision has 316 false positives and Naïve Bayes only 145 which means that Tree Decision can not predict in a proper way positive instances. It can be seen in **Figure 9** and also in **Table 4**, Naïve Bayes is the classifier that has more correct predictions, with 750 versus the second one, Perceptron, with 736.

The classifier with the worst AUC value is Random Forest with 0.70 AUC but Tree Decisions is the worst method taking into account the number of correct predictions, 668. By the other hand, MLP provides us with an AUC of 0.84 which means a great advantage over machine learning methods.

Table 4: Confusion Matrix per Classifier

Classifier	C	I	TP	FN	TN	FP
Naïve Bayes	**750**	250	355	105	395	145
SVM	733	267	346	113	387	154
Logistic Regression	741	259	351	110	390	149
Tree Decision	668	332	184	16	484	**316**
Random Forest	696	304	304	392	108	196
Perceptron	736	264	392	156	344	108

Although the best AUC comes from the Multilayer Perceptron, we take into account other parameters such as the training time or the optimal hiperparameters fitting, thus, based on all these factors we chose Naïve Bayes as the best classifier.

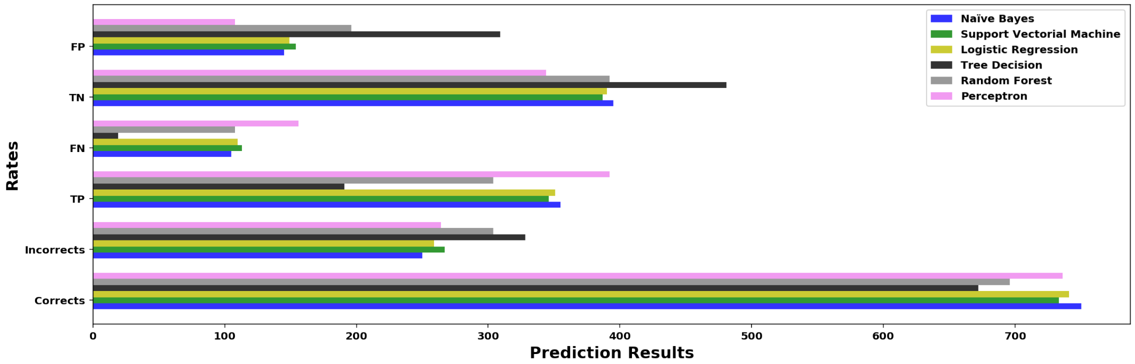
Once we have compared the main supervised machine learning methods tested in this experiment and based on the results that we have obtained and can be seen in **Figure 9**, we performed a quick test using the best classifier with tweets downloaded from the API REST. In this case, the tested algorithm was Naïve Bayes.

Table 5: Twitter Real Application

Sentiment Predicted	
Tweets	**Labels**
Have a great day as well! We have many more promos planned. Stay tuned!	1
I would be more than happy to further investigate this transaction.	0
That's AMAZING! Congrats on the 20th year. We greatly appreciate you as a member.	1

As we can observe in **Table 5**, tweets are indeed classified correctly. As we do not have **ground truth** and therefore no metric, we observe that the Naïve Bayes model trained in this study works with new inputs.

Figure 9: Results



5 CONCLUSIONS

Sentiment analysis is about understanding how humans are feeling related to an action, what could give to the companies with an online presence the power to understand the opinions of their customers online.

In this study we compared several supervised learning methods based on words frequencies in order to identify the most accurate model to analyse sentiment.

In **Table 3**, we can observe the main metrics which measure each model. In red, we can see that the lowest AUC is Random Forest and the highest in the Multilayer Perceptron. Also, it can be observed that Tree Decision has 0.76 in AUC but when we see other metrics we can see that precision metric is the weakest.

In conclusion, we can determine that the best machine learning classifiers in supervised learning to carry out this sentiment analysis studies are Naïve Bayes and Perceptron. Neural Networks, although the processing, training and fitting of hyperparameters are more computational expensive, provide us a precise binary prediction.

REFERENCES

[1] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Support vector machines for multiple-instance learning. In *Advances in neural information processing systems*. 577–584.

[2] Erik Cambria, Björn Schuller, Yunqing Xia, and Catherine Havasi. 2013. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent systems* 28, 2 (2013), 15–21.

[3] Koyel Chakraborty, Siddhartha Bhattacharyya, Rajib Bag, and Aboul Ella Hassanien. 2018. Comparative sentiment analysis on a set of movie reviews using deep learning approach. In *International Conference on Advanced Machine Learning Technologies and Applications*. Springer, 311–318.

[4] Nadia FF Da Silva, Eduardo R Hruschka, and Estevam R Hruschka Jr. 2014. Tweet sentiment analysis with classifier ensembles. *Decision Support Systems* 66 (2014), 170–179.

[5] Bijoyan Das and Sarit Chakraborty. 2018. An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation. *arXiv preprint arXiv:1806.06407* (2018).

[6] Kotzias et. al. 2015. UCI Machine Learning Repository. (2015). http://archive.ics.uci.edu/ml

[7] Simran Fitzgerald, George Mathews, Colin Morris, and Oles Zhulyn. 2012. Using NLP techniques for file fragment classification. *Digital Investigation* 9 (2012), S44–S49.

[8] Peter A Flach. 2003. The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In *Proceedings of the 20th international conference on machine learning (ICML-03)*. 194–201.

[9] Semuel Franko and Ismail Burak Parlak. 2018. A comparative approach for multiclass text analysis. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*. IEEE, 1–6.

[10] Jordi Gironés et al. [n. d.]. Minería de datos: modelos y algoritmos, Editorial UOC, 2017. *ProQuest Ebook Central* ([n. d.]).

[11] David W Hosmer, Trina Hosmer, Saskia Le Cessie, and Stanley Lemeshow. 1997. A comparison of goodness-of-fit tests for the logistic regression model. *Statistics in medicine* 16, 9 (1997), 965–980.

[12] Fauzia Idrees, Muttukrishnan Rajarajan, Mauro Conti, Thomas M Chen, and Yogachandran Rahulamathavan. 2017. PIndroid: A novel Android malware detection system using ensemble learning methods. *Computers & Security* 68 (2017), 36–46.

[13] Thorsten Joachims. 1996. *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization.* Technical Report. Carnegie-mellon univ pittsburgh pa dept of computer science.

[14] Monisha Kanakaraj and Ram Mohana Reddy Guddeti. 2015. Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques. In *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*. IEEE, 169–170.

[15] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 597–606.

[16] Namyeon Lee, Eunji Kim, and Ohbyung Kwon. 2018. Combining TF-IDF and LDA to generate flexible communication for recommendation services by a humanoid robot. *Multimedia Tools and Applications* 77, 4 (2018), 5043–5058.

[17] R Manikandan and R Sivakumar. 2018. Machine learning algorithms for text-documents classification: A review. *Machine learning* 3, 2 (2018).

[18] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242* (2013).

[19] Antonio Moreno and Teófilo Redondo. 2016. Text analytics: the convergence of big data and artificial intelligence. *IJIMAI* 3, 6 (2016), 57–64.

[20] S Fouzia Sayeedunnisa, Nagaratna P Hegde, and Khaleel Ur Rahman Khan. 2018. Wilcoxon Signed Rank Based Feature Selection for Sentiment Classification. In *Proceedings of the Second International Conference on Computational Intelligence and Informatics*. Springer, 293–310.

[21] Pravesh Kumar Singh and Mohd Shahid Husain. 2014. Methodological study of opinion mining and sentiment analysis techniques. *International Journal on Soft Computing* 5, 1 (2014), 11.

[22] Wen Zeng, Xiang Li, and Hui Li. 2018. Study on Chinese Term Extraction Method Based on Machine Learning. In *International Conference of Pioneering Computer Scientists, Engineers and Educators*. Springer, 128–135.

[23] Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1253.

Transfer Learning for Malware Multi-Classification

Mohamad Al Kadri, Mohamed Nassar, Haidar Safa
Computer Science Department
Faculty of Arts and Sciences
American University of Beirut (AUB)
[mga35|mn115|hs33]@aub.edu.lb

ABSTRACT

In this paper, we build on top of the MalConv neural networks learning architecture which was initially designed for malware/benign classification. We evaluate the transfer learning of MalConv for malware multi-class classification by extending its contribution in several directions: (1) We assess MalConv performance on a multi-classification problem using a new dataset composed of solely malware samples belonging to different malware families, (2) we evaluate MalConv on the raw bytes data as well as on the opcodes extracted from the reversed assembly samples and compare the results, (3) we validate the MalConv findings about regularization, and (4) we study MalConv performance when using a medium size dataset and limited computational resources and GPU. The obtained results show that MalConv performs equally well for multi-classification and its performance on raw byte sequences is comparable to opcodes sequences. DeCov regularization is shown to improve the accuracy results better than other regularization techniques.

CCS CONCEPTS

• **Security and privacy → Malware and its mitigation**; • **Computing methodologies → Machine learning**; **Neural networks**;

KEYWORDS

Transfer Learning, Malware, Classification, Regularization, Deep Learning

ACM Reference Format:
Mohamad Al Kadri, Mohamed Nassar, Haidar Safa. 2019. Transfer Learning for Malware Multi-Classification . In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3331076.3331111

1 INTRODUCTION

Malware data are fundamentally different than text and image data. For instance, a byte in a malware executable has different meanings depending on its location in the code and its context. It can be an instruction, a part of an instruction, a part of an address, an argument, a data item, etc. In contrast, a byte in an image or a video always represents the pixel intensity or the color code. A byte in a text sequence would always represent a letter. Therefore, dealing with malware data, executables, or software in general, requires different machine learning techniques and artifacts. This same idea has been highlighted in [25] and it is part of the motivation behind many research work on the topic of malware detection and classification using deep learning.

Raffed et. al. proposed MalConv as a convolutional neural network architecture for malware detection by eating a whole exe [18]. Previous machine learning work was based to a large extent on the quality of features extracted from the raw byte data. As a major shift, deep learning nowadays claims that it can automatically discover and encode relevant features without recurring to any tedious manual engineering and expert selection of features. Still, good architecture design is required to achieve high performance. MalConv [18] has been specially designed as a shallow model to deal with very long sequences (often millions of bytes). Deep learning architectures for sequences such as Recurrent Neural Networks (RNN) [23] and Long Short-Term Memory (LSTM) [9] usually deal with very small sequences representing sentences which makes it possible to go deeper with many layers. In MalConv, the whole malware byte code is represented as one sequence. The large length of sequences requires a design with a moderate number of layers to meet the available computational and memory resources.

In machine learning, transfer learning is to take knowledge the neural network has learned from one task and apply that knowledge as the starting point for training a model on a separate task. Transfer learning is usually successful when low-level features from the first task could help learn the second task. Two examples are learning to classify radiology images based on an object recognition classifier, or learning to drive quadcopters based on a self-driving car model. Transfer learning has been most useful where relatively little data are available to train a model for the target task.

In this paper, we propose transfer learning of the MalConv architecture and experiment with a new dataset. MalConv was initially designed for malware/benign classification and has not been tested in multi-class settings. We evaluate MalConv for malware multi-class classification and extend its contribution in several directions:

- We assess MalConv performance on a multi-classification problem using a new dataset composed of solely malware samples belonging to different malware families.
- We evaluate MalConv on the raw bytes data as well as on the opcodes extracted from the reassembled samples and compare the results.
- We validate the MalConv findings about regularization, especially that DeCov [4] is a much better regularizer than batch normalization [10]. The reason is that, as we will show, the

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06.
https://doi.org/10.1145/3331076.3331111

activation distributions are multi-modal and far from fitting a Gaussian distribution.

- We study the performance of MalConv with a medium size dataset and limited computational resources and GPU. We evaluate the technical limits of operating on a moderate GPU and a single off-the-shelf machine.

Note that since we do not have the original MalConv pre-trained model, we could not experiment with only replacing and re-training the last one or two layers. Instead, we have trained all the layers of MalConv using several parameterizations. We consider transfer learning because our data-set is very small compared to the datasets used to train the original MalConv. To our knowledge, this is the first work addressing transfer learning from malware/benign classification to malware multi-classification.

Our results show that MalConv performs equally for multi-classification and its performance on raw byte sequences is comparable to opcode sequences. DeCov regularization improves the accuracy results better than other regularization techniques.

The rest of this paper is organized as follows: We discuss related work in Section II. In Section III we present the MalConv architecture for multi-classification. In Section IV we introduce the dataset, perform experiments and discuss the obtained results. Section V concludes the paper and discusses future work.

2 RELATED WORK

Malware is a major threat to the Internet of today. Malware data are fundamentally different than text and images [25]. Accuracy numbers on malware clustering and classification are not representative and sometimes misleading. In [14] six commercial anti-viruses are shown to be biased and not better than a simple plagiarism detection algorithm. This bias is mainly due to unbalanced datasets where most malware instances are easy to classify. Machine learning algorithms in general and neural networks, in particular, are proposed as a way to cope with these challenges.

Neural networks have been around for decades. They resurged thanks to the unprecedented data availability and computational scale, and have achieved major breakthroughs in many domains such as games, visual object recognition, language modeling, and speech recognition. Deep learning is informally known as a set of recent neural network designs such as word embedding, convolutional networks, and recurrent/recursive networks. Deep learning has been proposed to enhance the two branches of malware analysis, namely static analysis, and dynamic analysis.

Static analysis is about extracting syntactical and semantical features from the binary or the disassembled malware using tools such as IDA [6]. Its main challenge is code obfuscation, metamorphic and polymorphic malware. Static analysis can be reinforced by deep learning as proposed in [22]. In that paper, the identification of function starts and ends in the binary code is addressed. Experiments with recurrent neural networks show that functions in binaries can be identified with greater accuracy and efficiency than many other machine learning algorithms.

Dynamic analysis runs the malware in a sandbox such as Cuckoo sandbox [21] and monitors its activities such as system calls and file access patterns. Dynamic analysis is known to be time and resource consuming. Moreover, its main challenge is that malware can detect the surrounding environment and keep calm [7]. Dynamic analysis can be reinforced by deep learning as proposed in [13]. This approach suggests classifying malware samples based on their system calls sequences. Experiments were performed with one-dimensional convolutional networks taking sequences of system calls in the form of a set of n-grams. However, the drawback of convolutional networks is that they do not explicitly model the sequential position of system calls. On the other hand, recurrent networks train a stateful model by using full sequential information. The drawback is that RNNs are more complex and more difficult to train. The authors show that by combining those two layers within the same hierarchy, malware detection capabilities are increased.

The malware dataset [20] that we intend to use in this work was used in much related work such as in [8]. In [8], two approaches were proposed: the first approach represents malware samples as 32 x 32 gray-scale images and inputs them to a convolutional network with max pooling followed by fully connected layers. The original idea of representing malware as an image has appeared in [17]. This approach suggests transforming the binary into a vector of 8-bit integers, which can be reshaped into a matrix and therefore viewed as a gray-scale image. However as discussed in MalConv [18], the receptive field of a convolution represents discontinuous sequences in the original malware. This fact suggests that using one-dimensional convolutional networks is more valuable. The second approach in [8] recognizes this fact and uses a scheme that initially appeared in [12]. The initial approach deploys a convolutional layer with multiple filter widths and feature maps on top of word vectors obtained using Word2Vec [15]. We report a very comparable performance based on transfer learning of MalConv in this paper.

A subset of the authors has previously proposed modeling malware as a language and experimented with a document-distance approach in [2]. Our approach showed promising preliminary results, but it still requires computational performance improvement. We also experimented with using t-SNE to throttle malware families in 2D or 3D for visualization purposes based on n-grams features [16].

Few work addresses transfer learning for malicious software classification. In [19], a malware family classification approach is presented based on the ResNet-50 architecture, which is a deep learning model for image classification. Therefore, the Malware samples were represented as byte grayscale images. However, representing malware as a grayscale image is lossy in terms of sequential information and has been criticized in literature. Transfer learning is also used in a limited way in [11]. In that work, a generative adversarial network (GAN) composed of a generator and a discriminator is proposed. The generator learns to produce fake malware samples and make them indistinguishable from real ones. The goal is to generate samples which are most similar to zero-day attacks. Transfer learning was needed to stabilize the generator based on a pre-trained auto-encoder of malware characteristics. The discriminator is supposed to detect zero-day attacks efficiently by learning to separate fake malware samples from real ones.

3 MALCONV ARCHITECTURE

MalConv [18] was originally designed for the task of malware/benign classification on raw bytes data. It has outperformed many other

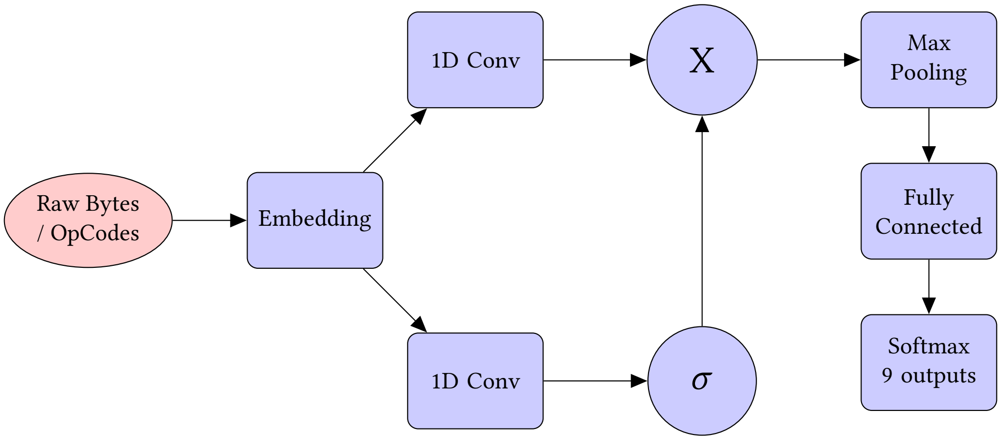


Figure 1: High-Level Diagram of the MalConv Architecture

Table 1: Distribution of samples among the families

Class	Family	Type	Nb. Of Instances	Percentage (%)
1	Ramnit	Worm	1541	14.20
2	Lollipop	Adware	2478	22.80
3	Kelihos_ver 3	Backdoor	2942	27.07
4	Vundo	Trojan	475	4.37
5	Simda	Backdoor	42	0.39
6	Tracur	Trojan Downloader	751	6.91
7	Kelihos_ver 1	Backdoor	398	3.66
8	Obfuscator.ACY	Obfuscated malware	1228	11.30
9	Gatak	Backdoor	1013	9.32

architectures including deep convolutional and various RNN with different attention models. MalConv is based on the idea of gated convolutional networks that was originally proposed in [5]. MalConv design is mainly driven by the following principles:

- Preserving a high level of generalization to previously unseen samples. This constraint has ruled out approaches that do not have convolution filters at all, since big parts of the executable may look benign while a small part somewhere in the file is malicious.
- Dealing with very large sequences. This constraint has limited the number of convolutional layers to keep up with memory resources.
- Dealing with information sparsity by applying max-pooling instead of average-pooling. Average-pooling may lead to loss of sparse features having high pitched responses.
- Avoiding applying RNN on top of CNN layers. It seems that RNN are overfitting the sequencing patterns at the output of the CNN and not generalizing well.

In this paper we propose transfer learning of MalConv to fit the multi-class malware classification settings as shown in Figure 1. It starts by feeding the bytes to an embedding layer (trainable lookup table with eight output dimensions). Otherwise, bytes would be considered similar if they have close numerical values, which is wrong. The embedding layer output is fed into two 1-D convolutional layers in parallel; only one of them has non-linear (sigmoid) activation. The convolutional layers have 128 filters (in depth) with rather a large filter width of 500 bytes combined with an aggressive stride of 500 bytes. The outputs of the two layers are elementwise multiplied and optionally passed to a rectified linear unit (ReLU). Then a temporal max pooling layer takes the global maximum of each of the 128 channels. The last part is a fully connected neural network (with optional ReLU activation) having 128 input nodes and nine softmax output nodes corresponding to the different malicious classes of malware. The classes will be presented in the experiments section.

An important factor in MalConv design is the regularization. The authors of MalConv suggested that penalizing the correlation between hidden state activations at the fully connected layer, as described in [4], to be the most effective form of regularization. Quoting from [3]: "Intriguingly, the commonly used batch-normalization actually prevented the models from converging and generalizing." In our experiments, We have obtained very similar results that will be discussed in the next section.

4 EXPERIMENTS

In this section, we present the results of the experiments on the multi-class dataset [20]. We measure the performance in terms of accuracy and loss. The accuracy is simply the ratio of the number of correct classifications to the number of all decisions made. The loss is defined as the cross-entropy function between the correct labels and the probabilities output by the network. The loss function is seen as a stronger indication of performance than mere accuracy. We used Keras [1] for the implementation of the tested networks. The main hardware component is a GeForce GTX 750 model with 2 GB Memory, and 5.0 compute capability. We start by presenting the dataset in the next paragraph.

4.1 Dataset

In this work, we used the data set provided by Microsoft and hosted at Kaggle [20]. The data set includes 10868 labeled samples and 10873 unlabeled ones. For each sample, the raw data and the meta-data are provided. The raw data contains the hexadecimal representation of the malware executable byte code, with the Portable Executable (PE) header removed to ensure sterility. The meta-data file is a manifest generated using the IDA disassembler tool. It represents the disassembled file (in X86 assembly language) containing various meta-data information such as function calls, strings, etc. The dataset contains malware samples belonging to the following nine families: Ramnit, Lollipop, Kelihos Ver. 3, Vundo, Simda, Tracur, Kelihos Ver. 1, Obfuscator.ACY and Gatak. One challenge of this data set is the unbalanced sizes of different families. The distribution of instances for the training dataset is shown in Table 1. Classes 4, 5, 6 and 7 are underrepresented as compared to the other families.

MalConv models a malware sample as one long sequence of bytes. Also, We experiment with another model which is one long sequence of opcodes. The sequence of opcodes is on average 60 times shorter than the sequence of bytes. Therefore using opcodes requires fewer GPU resources. Still, a rather costly preprocessing phase of reassembling is required. Sometimes reassembling is erroneous when malware authors have protection techniques against some known disassemblers. For instance in our dataset, 4 Kelihos_ver3, 22 Vundo, 6 Kelihos_ver1, 2 Lollipop, and 9 Obfuscator.ACY samples do not have any opcode.

Using sequences of raw bytes is more attractive since it has all the information. However, it can contain a lot of noise which is deliberately injected by the malicious developers.

In all cases, the trade-off between using sequences of bytes or sequences of opcodes seems interesting. Our intuition for opcodes is that the obfuscation techniques used by the malware authors introduce most of the time junk assembly instructions and a different context for each instruction. If malware families use different obfuscation techniques leading to different opcode sequencing, then this can be encoded by embedding layers and caught by convolutional layers as features. We test this hypothesis by embedding the opcodes of each malware family in our data set using Word2Vec [15]. The embedding is presumably sensitive to the absence and presence of some opcodes, different ordering and different surroundings also referred to as a context.

Figure 2 illustrates how malware classes have different embedding representations. Each subfigure represents the embedding of the union set of all opcodes for the samples of a given malware class. The embedding is originally learned by a Word2Vec model with a context window size of 5 and eight output dimensions. The eight dimensions are then compressed into two using t-SNE dimensionality reduction with perplexity of 40 and PCA initialization.

Some special opcodes are cross marked in the scatter plots. What is interesting in this kind of plots is not the absolute position of an opcode in the space but rather the relative neighboring information. We notice, for example, that `mov` and `jmp` are very close in class 1 whereas they are rather separated in class 2.

4.2 Bytes vs. Opcodes

We split the labeled data into three sets:

- training set: 6526 samples (60%)
- validation set: 2175 samples (20%)
- testing set: 2175 samples (20%)

We show in Figure 3 the per-class validation and testing accuracy for the byte sequences and for the opcode sequences.

The two models show good yet similar performance. In exception, the worst accuracy is for class 5 (Simda). This is due to having very few samples available for training. We also tested using the unlabeled data sets and obtained an overall log loss of 0.093 for opcodes and 0.122 for raw bytes as per returned Kaggle private scores.

4.3 Regularization

We start by showing that using batch normalization [10] on top of the convolutional filters is mostly not a good regularizer for our dataset.

Figure 4 evaluates the validation accuracy for raw bytes and opcodes in terms of training epochs for small batch sizes (only four sequences for both bytes and opcodes). It shows that using batch normalization has worse results in both cases. Still training for more epochs seems beneficial in the case of opcodes.

Figure 5 evaluates the validation accuracy for raw bytes and opcodes in terms of training epochs for large batch sizes (16 sequences for bytes and 128 sequences for opcodes). Similarly, using batch normalization has worse results in both cases. Batch normalization seems indifferent in the case of bytes and much worse in the case of opcodes in this experiment.

So why is batch normalization, which is known to be a very successful technique, spectacularly failing in our case? It turns out that batch normalization works better when the output of the convolution filters has a Gaussian distribution. This does not seem the case for our data which is showing multi-modal distributions. As an example, we plot the probability density function of an early activation node in the network along with the PDF of a fitting Gaussian distribution in Figure 6.

MalConv authors propose to use DeCov [4] as a better regularizer. DeCov explicitly penalizes the correlation between the activations in the fully connected layers, hence preventing these and the previous convolutional layers from overfitting and redundantly encoding the same information. It does so by directly adding the correlation terms to the loss function. Therefore, DeCov works in a very similar way to dropout regularization [24] except that the way dropout works is much more implicit.

Table 2 shows that DeCov has the same effect on our dataset, both for byte and opcode sequencing models. DeCov has even better

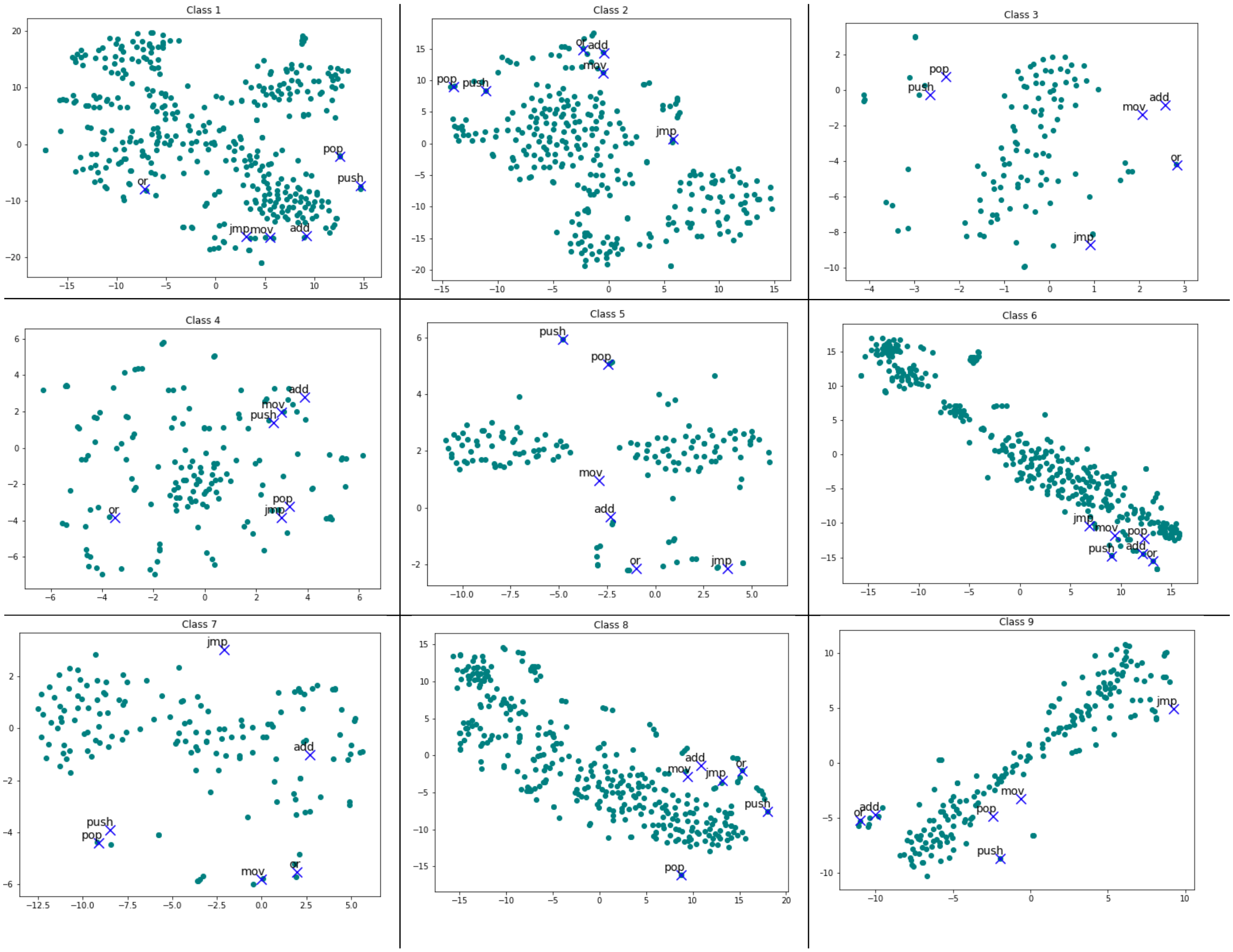


Figure 2: Embedding representation for each malware class

Table 2: Classification Accuracy and Cross Entropy (training/testing) for Bytes and Opcodes without DeCov, with DeCov and with Dropout (50%)

	MalConv without DeCov		MalConv with DeCov		MalConv with Dropout	
Test/Validation Set	Accuracy	Cross-Entropy	Accuracy	Cross-Entropy	Accuracy	Cross-Entropy
Opcodes	95/95%	0.30/0.26	96/96%	0.21/0.20	93/93%	0.44/0.44
Raw Bytes	95/95%	0.33/0.24	97/98%	0.22/0.13	95/95%	0.20/0.19

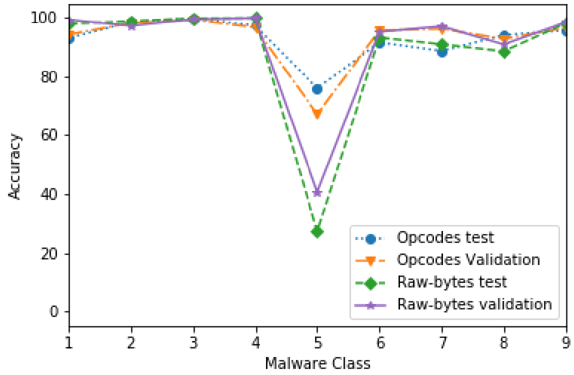


Figure 3: Average classification accuracy per malware class for the raw bytes and opcodes data.

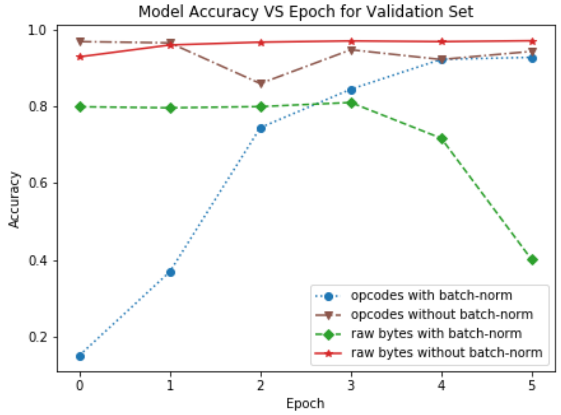


Figure 4: Comparison of the validation set accuracy of with-vs.-without Batch Normalization for small batch sizes

performance than dropout. Dropout with a ratio of 50% has better results on bytes than on opcodes. However, tuning the ratio of dropout (e.g., by decreasing the drop rate to 25%) leads to better results.

4.4 GPU Memory limitations and sequence length vs. batch size trade-off

Our experiments were constrained by using only one GPU GeForce GTX 750, 2 GB memory. This limitation has impacted the length of sequences that can be dealt with. We have tried to surmount this problem by decreasing the batch size, changing the network parameters at different layers and truncating the sequences. This has led to three network configurations:

- *Network1* (convolution stride = 500, filter width = 500, number of filters = 128),
- *Network2* (convolution stride = 1000, filter width = 1000, number of filters = 64),

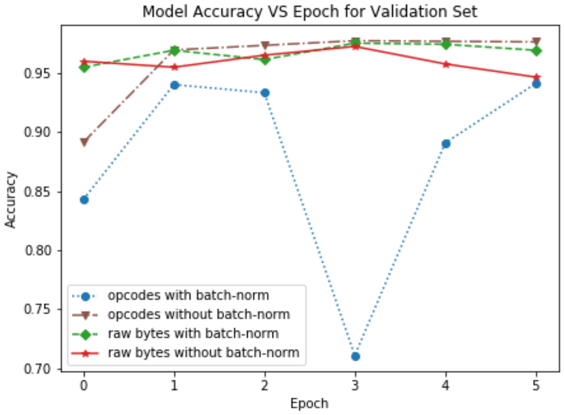


Figure 5: Comparison of the validation set accuracy per epoch of with-vs.-without Batch Normalization for large batch sizes

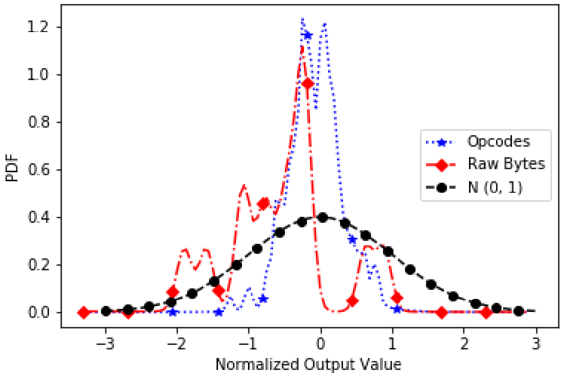


Figure 6: The probability density functions of a convolutional filter output in case of opcodes vs. bytes vs. Gaussian

- and *Network3* (convolution stride = 100, filter width = 500, number of filters = 160).

Network1 is very similar to the original MalConv architecture, *Network2* is less memory demanding and *Network3* is more memory demanding. In Table 3, we list the maximum possible sequence length for each network type and network parameters. We obtained these values using an iterative binary search. Generally increasing the batch size and the number of filters improves the classification accuracy even when sequences are moderately truncated.

5 CONCLUSION

In this paper, we have assessed the transfer learning of the MalConv neural networks architecture for a multi-class malware classification problem, which is a shift from the original goal of the MalConv design. However, we found that MalConv has very similar performance on the studied dataset and that the lessons learned by

Table 3: Maximum Possible Sequence Length under Different Batch sizes and Network parameters

Data type (Embedding size) →	Opcodes (8 dimensions)		Bytes (4 dimensions)	
Network type ↓	Batch size	Maximum length	Batch size	Maximum Length
Network1	64	150,247	8	2,342,505
	128	64,391	16	1,171,252
	256	30,049		
Network2	64	163,125	8	2,459,630
	128	85,855	16	1,210,294
	256	34,342		
Network3	64	128,783	8	1,756,879
	128	62,245	16	976,044
	256	27,903		

MalConv are still valid for the case of malware multi-classification problem. Another deviation that we have experimented with is considering the malware as a sequence of opcodes rather than a sequence of raw bytes. Although this approach requires preprocessing overhead, it makes the length of the sequences much more manageable given limited GPU memory resources. In particular, we validate the fact that regularizers such as dropout and DeCov are much more likely to improve the network convergence whereas batch normalization can have negative effects. To our knowledge, this is the first attempt to apply transfer learning from malware/benign classification to malware multi-classification.

The size of the target dataset is rather small compared to the datasets studied in the MalConv paper, which is a strong motivation for transfer learning. In future work, we aim to experiment with more datasets derived from different sources to validate the viability of transfer learning in malware data on a larger scale. We also would like to experiment transfer learning using the original pre-trained model of MalConv from NVIDIA, if publicly available.

ACKNOWLEDGMENT

This work was supported partially by a grant from the university research board of the American university of Beirut (URB-AUB-2018/19).

REFERENCES

[1] Keras: The python deep learning library. https://keras.io/. Accessed: 2018-07-14.

[2] Y. Awad, M. Nassar, and H. Safa. Modeling malware as a language. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[3] J. Barker. Malware detection in executables using neural networks. https://devblogs.nvidia.com/malware-detection-neural-networks/. Accessed: 2018-07-14.

[4] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

[5] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, 2016.

[6] C. Eagle. *The IDA pro book.* No Starch Press, 2011.

[7] O. Ferrand. How to detect the cuckoo sandbox and to strengthen it? *Journal of Computer Virology and Hacking Techniques*, 11(1):51–58, 2015.

[8] D. Gibert Llauradó. Convolutional neural networks for malware classification. Master's thesis, Universitat Politècnica de Catalunya, 2016.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[11] J.-Y. Kim, S.-J. Bu, and S.-B. Cho. Malware detection using deep transferred generative adversarial networks. In *International Conference on Neural Information Processing*, pages 556–564. Springer, 2017.

[12] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[13] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert. Deep learning for classification of malware system call sequences. In *Australasian Joint Conference on Artificial Intelligence*, pages 137–149. Springer, 2016.

[14] P. Li, L. Liu, D. Gao, and M. K. Reiter. On challenges in evaluating malware clustering. In *International Workshop on Recent Advances in Intrusion Detection*, pages 238–255. Springer, 2010.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[16] M. Nassar and H. Safa. Throttling malware families in 2d. *arXiv preprint arXiv:1901.10590*, 2019.

[17] L. Nataraj, S. Karthikeyan, G. Jacob, and B. Manjunath. Malware images: visualization and automatic classification. In *Proceedings of the 8th international symposium on visualization for cyber security*, page 4. ACM, 2011.

[18] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas. Malware detection by eating a whole exe. *arXiv preprint arXiv:1710.09435*, 2017.

[19] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. De Geus. Malicious software classification using transfer learning of resnet-50 deep neural network. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1011–1014. IEEE, 2017.

[20] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi. Microsoft malware classification challenge. *arXiv preprint arXiv:1802.10135*, 2018.

[21] C. Sandbox. Automated malware analysis. *https://cuckoosandbox.org*, 2013.

[22] E. C. R. Shin, D. Song, and R. Moazzezi. Recognizing functions in binaries with neural networks. In *USENIX Security Symposium*, pages 611–626, 2015.

[23] H. T. Siegelmann and E. D. Sontag. On the computational power of neural nets. *Journal of computer and system sciences*, 50(1):132–150, 1995.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[25] J. Sylvester. Malconv: Lessons learned from deep learning on executables. http://www.jsylvest.com/blog/2017/12/malconv/. Accessed: 2018-07-14.

Blockchain-based Risk Mitigation for Invoice Financing

Meriem Guerar, Luca Verderame, Alessio Merlo
DIBRIS - University of Genoa
Genoa, Italy
name.surname@unige.it

Mauro Migliardi
DEI - University of Padua
Padua, Italy
mauro.migliardi@unipd.it

ABSTRACT

The market for *invoice financing* has been steadily growing in the last few years and has been the third financing market in size in 2016. Most solutions in this field are based on private platforms and even the new proposals based on blockchain are mostly adopting a private, permissioned blockchain. In this paper, we propose an idea based on a public blockchain that allows both fully open and group-restricted auctioning of invoices. Furthermore, our proposal introduces a reputation system that is based on the past behavior of entities, as it is photographed by the public blockchain, to allow insurance companies modulate the cost of the insurance contracts they offer. This combination guarantees the complete transparency and tamperproof-ness of a public blockchain, while it allows reducing insurance costs and fraud possibilities.

CCS CONCEPTS

• **Security and privacy → Database and storage security**; **Database activity monitoring**.

KEYWORDS

Blockchain, Ethereum, Smart contract, Auction, Invoice factoring, IPFS

ACM Reference Format:
Meriem Guerar, Luca Verderame, Alessio Merlo and Mauro Migliardi. 2019. Blockchain-based Risk Mitigation for Invoice Financing. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3331076.3331093

1 INTRODUCTION

Companies work hard to ensure economic liquidity and maintain steady cash-flow, that said, those important factors are seriously affected by the long invoicing due dates which represent a big challenge, especially for small to medium enterprises (SMEs). In order to overcome this issue companies make use of different forms of invoice financing such as factoring. This type of financing enables businesses to cash-in invoices before their due date. The process of factoring can be described as follows: an SME sells the invoice to a factoring company which is often a financial institution for a pre-agreed percentage of the invoice amount, the buyer then pays the factoring company the full invoice amount on the due date. While this helps the SME solve the cash-flow issues, it exposes the factoring companies to serious fraud risks mainly because of the lack of communication among themselves. In fact, a well known fraud risk in factoring is double financing, where the SME sells the same invoice to more than one financial institution. The buyer will naturally pay the invoice once, paying only one institution and leaving the rest unpaid. Another considerable risk is represented by a situation where the buyer refuses to pay as agreed on the due date of the invoice. One of the main reasons that leads to this is the fact that a financial institution does not have a direct relationship with the buyer and relies only on the information provided by the seller, in our example the SME.

One potential solution to the double financing problem is an invoice financing platform hosted on a centralized database where all the potential invoice-buyers can verify whether the invoice has been already funded or is still available. However, centralized systems can be expensive, they are a single point of failure, and they are prone to privacy infringement, data manipulation and attacks which may make them unreliable and untrustworthy. Luckily, with the emergence of blockchain technology and smart contracts, we no longer have to rely on centralized systems. Blockchain may be used to implement an immutable, trusted, and decentralized ledger [6] that relies on a consensus algorithm to decide which data is appended [13].

In this paper, we propose an invoice financing solution through auctioning based on InterPlanetary File System (IPFS) [2] and Ethereum blockchain [16]. The invoice data is stored on the IPFS while its corresponding IPFS hash is stored into a blockchain smart contract in order to ensure integrity, traceability and authenticity of the invoice. Moreover, the proposed solution uses a reputation system which contributes to reduce the fraud risks. The rest of this paper is structured as follows: In Section 2 we introduce the invoice financing solution; in Section 3 we describe the frauds scenario and countermeasures; in Section 4 we present related work. Finally, Section 5 concludes this paper.

2 THE PROPOSED INVOICE FINANCING SOLUTION

2.1 System overview

In this paper, we propose a prototype of an invoice financing platform for SME based on InterPlanary File System (i.e., IPFS), reputation profiles, and smart contracts hosted on Ethereum blockchain. Every function call that modifies the blockchain state or smart contract executed on the Ethereum blockchain requires Gas [1]. Gas is a unit that is used to calculate the amount of fees that need to be paid to the network in order to execute an operation. Since the

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331093

invoice data are very sensitive and storing this data directly in the blockchain is very expensive, we do not plan to store the whole invoice inside the blockchain. On the contrary, we propose to use IPFS to store these data in a decentralized, distributed manner that is publicly and globally accessible through the use of IPFS hashes. At the same time, to control access to the data, we encrypt the IPFS hash with the authorized investors public keys and store only these into a smart contract. Thus, any modification of the invoice content would change the IPFS hash, and would then not match the hash stored within the smart contract. The confidentiality of invoice data is ensured because only the authorized investors will be able to access it using their private keys.

The main components of our platform are:

- a smart contract hosted on the Ethereum blockchain,
- the Ethereum client,
- IPFS,
- a web app.

The web app provides a graphical user interface for the Ethereum client, which in turn interacts with the smart contract on the Ethereum blockchain. The roles of the participants can be summarized as follows:

Seller: is a company that has the goods to be packaged and transferred to the buyer and it is looking to improve its cash flow by creating a smart contract capable of selling the invoice to one of the investors enrolled in the platform through an auction. This kind of company is usually an SME.

Buyer: is a company that would like to purchase the goods from the seller by paying the shipping amount on delivery and benefits from the delayed payment of the full invoice amount (i.e., the price of goods plus taxes).

Authorized investor: is a person or a financial institution that is allowed to participate in the auction to buy the invoice at a price lower than its real value to gain a profit.

Insurance: is responsible to reimburse the authorized investor in case the buyer refuses to pay.

Unlike the traditional financing model, our platform does not limit the factoring service to banks and financial companies. Any investor can subscribe to the web app and make an offer to participate in the auction of an invoice. The highest offer made by an authorized investor that satisfies the minimum requested amount wins the auction once the bidding time has expired. This enables the SMEs to invite a large number of investors around the world and get the best financing offer in short time and with less effort to get funding.

At the same time, the buyer will benefit from the delayed invoice payment to optimize the use of their working capital.

2.2 Challenges

Since the investors do not have any direct knowledge of either the seller or the buyer, they are exposed to a considerable amount of risk. As an example, there is the risk of the invoice not being paid as agreed by the buyer; another significant risk is the seller knowingly submitting false, modified or duplicate invoices with the intent to commit a fraud, either acting alone or in collusion with the buyer. A solution might be to add risk insurance to refund the investor; however, in the absence of significant countermeasures aiming at reducing the fraud opportunity, the cost of such an insurance will make the whole operation economically unfeasible. Hence, the simple addition of an insurance is not considered a viable solution.

2.3 System design

The proposed platform mitigates these risks by adding transporter entity and reputation profile. The former provides information about shipping status while the latter shows the list of invoices that has been paid or unpaid by the buyer on the due date without showing the confidential data. This can help investors in the selection of trustworthy counterparts while pushing malicious buyers off the system.

The platform allows the seller and their counterparts to register by selecting the account type (e.g., seller account, investor account, etc) and providing an identity certificate which is unique to make sure that they can not create another account with a clean reputation profile in case of fraud. The services are provided according to the type of the account and every time the contract data changes, a notification is sent to the counterpart.

As shown in Figure 1, the seller writes the invoice data into IPFS and creates a smart contract that specifies the minimum amount required to participate in the auction and the hash to retrieve the invoice from IPFS. Then, he deploys it into the Ethereum blockchain. If the invoice is genuine, the buyer accepts the invoice and pays the shipping amount. When he accepts the invoice the buyer states that he verified all the information mentioned in the invoice and he agreed to pay the shipping amount immediately and the entire amount on the due date as specified in the invoice. Afterward, the investors can participate in the auction and thus read the invoice data and make an offer after checking the following conditions:

- the invoice has been accepted by the buyer;
- the "invoice ID" has not been submitted before;
- the buyer confirmed the delivery in order;
- the reputation profiles of both the seller and the buyer show that they are trustworthy.

If the reputation profile shows that one of them is untrustworthy or the invoice does not meet one of the above mentioned requirements, then it will not be funded by the investors. An investor that decides to finance an invoice in spite of the above mentioned problems is fully responsible of his decision and knows that, in case of fraud, his request of refund will be rejected by the insurance. Beside protection against double financing and submitting false or modified invoice, our platform mitigates the risk of a buyer that does not pay as agreed. In fact, in our platform the reputation profile will show that a buyer is untrustworthy and investor may freely take a fully informed decision if they want to run the risk. Thus, our platform facilitates the invoice financing for SME and reduces the risk of frauds.

2.4 The proposed invoice financing workflow

Figure 2 illustrates the message sequence diagram of selling the invoice through an auction with two possible scenarios. In the first scenario the buyer pays on due date of the invoice while in the second the buyer refuses to pay. The interactions between the different entities with the smart contract are as follows:

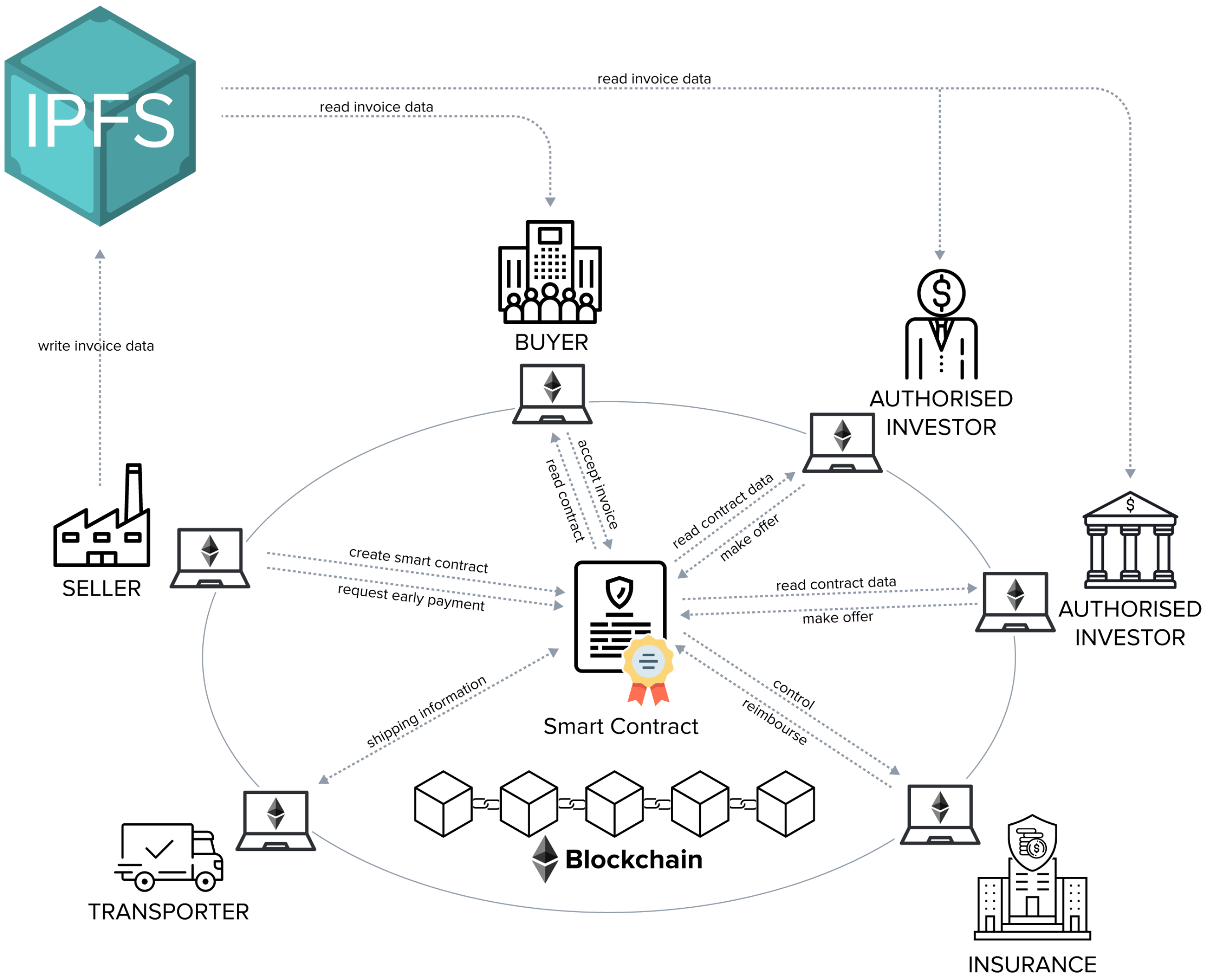


Figure 1: Invoice financing solution based on blockchain and IPFS.

(1) The seller creates a smart contract and deploys it in the Ethereum blockchain. The seller can choose to open the auction to all the investors in the platform or only to some predefined investors. In case of two authorized investors, the main contents of the smart contract are: hash (Invoice ID), shipping amount, the minimum bid requested, the highest bid, offers, auction deadline, shipment status and IPFS hash encrypted with public key of investor 1, 2 and the buyer.

(2) The buyer decrypts the IPFS hash using his private key and verifies the invoice data. If the invoice is genuine the buyer accepts the invoice and performs a safe payment of the shipping price. The smart contract holds this amount of Ether until the delivery.

(3) The transporter verifies if the invoice has been accepted by the buyer then, updates the shipment status on the smart contract to "in transit" upon receiving the goods.

(4) The buyer verifies if the shipment status on the smart contract is "in transit" then, updates it to "delivered" once the goods are received. The smart contract payout the transporter for the shipment.

(5) The investors verify the participation conditions mentioned above in order to decide whether to bid on this invoice or not.

(6) In case all the conditions are met, the first investor places his bid which should be higher than the minimum bid requested by the seller.

(7) The second investor places his bid which should be higher than the highest bid (i.e., bid 1). The highest bidder become the owner of the invoice when the auction ended.

(8) The seller asks for an early payment when the auction ended. The smart contract transfers the highest bid to the seller.

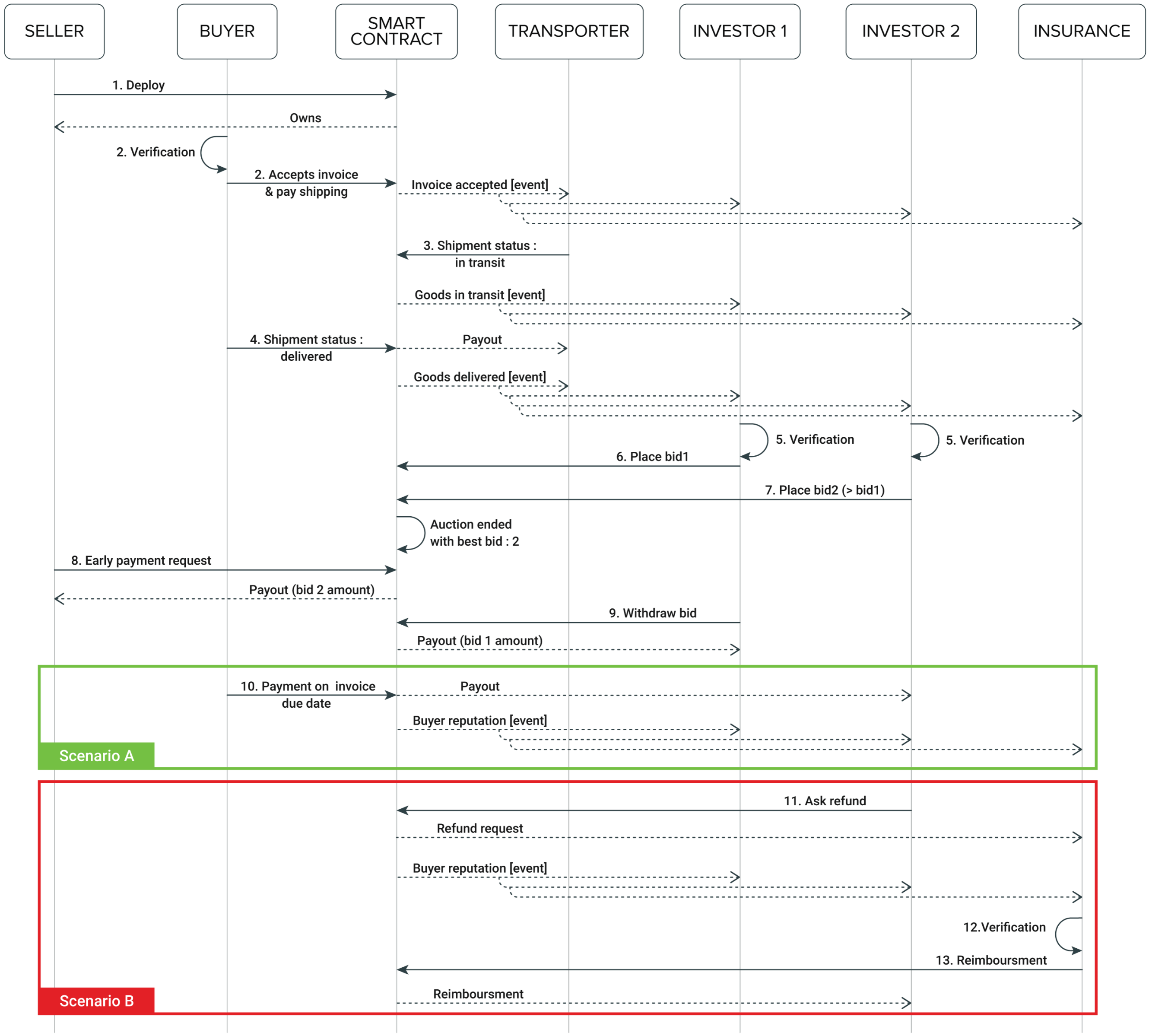


Figure 2: Sequence diagram of the proposed invoice financing workflow.

(9) When the auction ends, the investor 1 asks to withdraw his funds because he did not win the auction. The smart contract sends to the investor 1 his corresponding bid amount.

(10) **In scenario A**, the buyer pays the entire amount on due date of the invoice to investor 2 through the smart contract. An event ***BuyerReputation(BuyerAddress,"invoice paid on due date")*** will be triggered to help in tracing the buyer reputation and in notifying all parties.

(11) **In scenario B**, the buyer did not pay on due date of the invoice as agreed and thus investor 2 sends a refund request. Two events will be triggered ***RefundRequest(msg.sender, "Refund request")*** to notify the insurance and ***BuyerReputation(BuyerAddress, "Unpaid invoice on due date")*** to create notification and save a log about the buyer reputation. In this scenario, the buyer profile will show that this buyer is untrustworthy.

(12) The insurance verifies if the investor 2 did not ask refund before and he made the necessary verification before participating in the auction.

(13) The insurance refunds the investor 2 through the smart contract.

It is important to mention that in step 8, the seller manually invokes the smart contract when the auction ends to receive his money because the contract cannot activate itself; however, automating the reimbursement for investors that did not win the auction is possible by relying on step 9 on step 8. Nevertheless, we added step 9 to let the investors withdraw their funds rather than push funds to them automatically for the following security reasons: i) Sending ether back to all the investors that did not win auction could run out of gas. ii) Sending ether to unknown addresses could lead to security vulnerabilities [5].

3 FRAUD SCENARIOS AND COUNTERMEASURES

In this section, we present the possible fraud scenarios and we explain how the proposed solution, without relying on trusted third parties and just leveraging smart contracts and public blockchain technology, reduces the possibility of frauds in invoice financing between mutually untrusted entities.

All involved entities will be able to share and monitor the information related to invoice, auction, shipping and payment in a transparent manner. In addition, these information are immutable and cannot be changed. Therefore, the information that is used to build reputation profile is reliable.

Scenario 1: The seller knowingly submits a false or modified invoice. Our solution prevents this fraud because the invoice will not be funded by the investor if it has not been already accepted by the buyer. The buyer will be interested into accepting the invoice only if it is genuine because his reputation is at stake and he could lose the shipping amount.

Scenario 2: The buyer colludes with the seller, he accepts the false invoice submitted by the seller to commit a fraud and split with the seller the amount of Ether received from the investor. In this case, the buyer will be identified as untrustworthy. Furthermore, this is not enough to get funding, because the investor verifies also if the transporter receives the goods before deciding to finance the invoice.

Scenario 3: The seller submits a duplicate invoice in order to have double financing. Our platform enables both the buyer and the investor to verify that the invoice has not been submitted before because of the unique "Invoice ID" and the transparency guaranteed by the public blockchain.

Scenario 4: The buyer refuses to pay the investor in due time as stated on the invoice because he did not receive the goods. Our platform enables the investor to check if the goods has been delivered with a confirmation from the buyer before participating in the auction. The transporter will be interested into having the delivery confirmed by the buyer because his payment depends on the shipment status. Otherwise, the transporter will not accept to deliver the goods.

Scenario 5: The buyer receives the goods but refuse to pay on due date of the invoice. In this case, the investor will be refunded by the insurance and this buyer will be easily identified as malicious and untrustworthy through his reputation profile.

4 RELATED WORK

Most researchers, when proposing blockchain based solutions for invoice financing focus mostly on the issue of double financing.

Nijeholt et al. [9] proposed DecReg, a framework based on blockchain technology to address the "double-financing" issue in factoring. The framework has been implemented on a private blockchain. The access to the blockchain is controlled by a central authority (CA). Authors pointed out that the only feasible attack would be a collusion between the seller and the CA, where the CA prevents the financial institution from accessing the network which makes it vulnerable to double-financing. Hence, the financial institution should halt invoice financing until it regains access to the blockchain network.

Hofmann et al. [7] stated that the registration of invoice on the blockchain provides the opportunity to prevent fraud and double-financing issues in invoice discounting and factoring. Each invoice distributed across the network is hashed, timestamped, and given a unique identifier to prevent multiple financing on that particular invoice. However, authors did not provide implementation details such as whether the invoice is registered in public or private blockchain and how the different parties interact with each other.

Similarly, Nicoletti et al. [14] stated that blockchain can play an important role in preventing fraud during procurement finance solution implementation and notably reverse factoring. Blockchain provides complete traceability and real-time visibility on invoices status which prevent the fraudulent organizations from extracting funds from multiple financial institutions by using the same invoice.

In [15], authors proposed a conceptual framework based on blockchain technology for reverse factoring and dynamic discounting. Efficiency, transparency, and autonomy were identified as blockchain value drivers that will improve supply chain finance solutions.

Bogucharskov et al. [3] presented possible interaction between supplier, customer and factor in blockchain-based factoring application. In their interaction model, the factor provides funding to the supplier upon the confirmation of the customer that he received the goods. However, authors did not take in consideration the fraud risks if the supplier or costumer are untrustworthy or malicious. In addition to that storing invoice in the public blockchain is very expensive both from the storage and from the computational point of view.

Kayal et al. [8] stated that blockchain technology can be a powerful tool to tackle the financing problems of SMEs. In addition, they conducted an exploratory research into the appetite of the stakeholders involved in invoice factoring and inventory finance for adopting the blockchain technology.

5 CONCLUSION

In this paper we have put forward an idea for the invoice factoring and financing problem that is based on the IPFS, the Ethereum blockchain, smart contracts and reputation profiles. Our proposal is expected to provide a higher level of transparency than most solutions previously proposed, as it uses a public blockchain instead of a private one. Besides, the use of a proof-of-work based public blockchain also guarantees a better resilience to tampering and collusion. Finally, as we showed in this paper, our solution is capable of preventing most practical cases of frauds and, by providing better guarantees, it allows lowering the costs of insurance that is needed to protect the involved parties from residual fraud cases.

As a future extension, it is worth pointing out that, in principle, the adoption of a public blockchain based on proof-of-work may lead to energy wasting, as each fraud attempt carried out by any of the involved parties is expected to lead to some form of energy loss. To this aim, we argue that the energy impact of the adoption of a public blockchain in actual invoice financing scenarios should be investigated in future works, as well as energy-wasting related attack that malicious parties can willingly attempt. We plan to model the energy consumption of a public by leveraging models previously adopted in other contexts (like, e.g., [4, 10–12]).

REFERENCES

[1] [n. d.]. What is Gas? https://kb.myetherwallet.com/posts/transactions/what-is-gas/ Accessed: 2019-05-20.

[2] Juan Benet. 2014. IPFS - Content Addressed, Versioned, P2P File System. *CoRR* abs/1407.3561 (2014). arXiv:1407.3561 http://arxiv.org/abs/1407.3561

[3] A.V. Bogucharskov, I.E. Pokamestov, K.R. Adamova, and Zh.N. Tropina. 2018. Adoption of Blockchain Technology in Trade Finance Process. *Journal of Reviews on Global Economics* 7, 7 (nov 2018), 510–515. https://doi.org/10.6000/1929-7092.2018.07.47

[4] N. Gobbo, A. Merlo, and M. Migliardi. [n. d.]. A denial of service attack to GSM networks via attach procedure. 8128 LNCS ([n. d.]), 361–376. https://doi.org/10.1007/978-3-642-40588-4_25

[5] Neville Grech, Michael Kong, Anton Jurisevic, Lexi Brent, Bernhard Scholz, and Yannis Smaragdakis. 2018. MadMax: Surviving Out-of-gas Conditions in Ethereum Smart Contracts. *Proc. ACM Program. Lang.* 2, OOPSLA, Article 116 (Oct. 2018), 27 pages. https://doi.org/10.1145/3276486

[6] H. R. Hasan and K. Salah. 2018. Blockchain-Based Proof of Delivery of Physical Assets With Single and Multiple Transporters. *IEEE Access* 6 (2018), 46781–46793. https://doi.org/10.1109/ACCESS.2018.2866512

[7] Erik Hofmann, Urs Magnus Strewe, and Nicola Bosia. 2018. *Discussion—How Does the Full Potential of Blockchain Technology in Supply Chain Finance Look Like?* Springer International Publishing, Cham, 77–87. https://doi.org/10.1007/978-3-319-62371-9_6

[8] Alex Kayal, Jingwen Yao, Judith Redi, and Erich C.G. Schnoeckel. [n. d.]. *Financing Small & Medium Enterprises with Blockchain: An Exploratory Research of Stakeholders Attitudes.* Chapter Chapter 4, 65–83. https://doi.org/10.1142/97817863463910004 arXiv:https://www.worldscientific.com/doi/pdf/10.1142/9781786346391 0004

[9] Hidde Lycklama à Nijeholt, Joris Oudejans, and Zekeriya Erkin. 2017. DecReg: A Framework for Preventing Double-Financing Using Blockchain Technology. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts (BCC '17).* ACM, New York, NY, USA, 29–34. https://doi.org/10.1145/3055518.3055529

[10] A. Merlo, M. Migliardi, and P. Fontanelli. [n. d.]. Measuring and estimating power consumption in Android to support energy-based intrusion detection. 23, 5 ([n. d.]), 611–637. https://doi.org/10.3233/JCS-150530

[11] M. Migliardi and A. Merlo. [n. d.]. Energy consumption simulation of different distributed intrusion detection approaches. 1547–1552. https://doi.org/10.1109/WAINA.2013.214

[12] M. Migliardi and A. Merlo. [n. d.]. Modeling the energy consumption of distributed IDS: A step towards Green security. 1452–1457.

[13] Satoshi Nakamoto. 2009. Bitcoin: A peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf

[14] Bernardo Nicoletti. 2018. *Fintech and Procurement Finance 4.0.* Springer International Publishing, Cham, 155–248. https://doi.org/10.1007/978-3-030-02140-5_6

[15] Yaghoob Omran, Michael Henke, Roger Heines, and Erik Hofmann. 2017. Blockchain-driven supply chain finance: Towards a conceptual framework from a buyer perspective. In *IPSERA 2017.* Budapest - Balatonfüred, 1–15. https://www.alexandria.unisg.ch/251095/

[16] Gavin Wood. 2017. Ethereum: A secure decentralised generalised transaction ledger EIP-150 REVISION (759dccd - 2017-08-07). https://ethereum.github.io/yellowpaper/paper.pdf Accessed: 2018-01-03.

Database system comparison based on spatiotemporal functionality

Antonios Makris
Dept. of Informatics and Telematics,
Harokopio University of Athens
Athens, Greece
amakris@hua.gr

Konstantinos Tserpes
Dept. of Informatics and Telematics,
Harokopio University of Athens
Athens, Greece
tserpes@hua.gr

Dimosthenis Anagnostopoulos
Dept. of Informatics and Telematics,
Harokopio University of Athens
Athens, Greece
dimosthe@hua.gr

Mara Nikolaidou
Dept. of Informatics and Telematics,
Harokopio University of Athens
Athens, Greece
mara@hua.gr

Jose Antônio Fernandes de Macedo
Department of Computing, Federal
University of Ceará
Fortaleza, Brazil
jose.macedo@dc.ufc.br

ABSTRACT

The amount of sources and sheer volumes of spatiotemporal data have met an unprecedented growth during the last decade. As a consequence, a rapidly increasing number of applications are seeking to generate value by crunching those data. The development of a system that will tap into the potential value of the spatiotemporal big data analysis for a multitude of applications remains one of the biggest challenges in computer engineering. This paper delves into the key-characteristics of the most prominent suchlike systems. In particular, it provides a thorough analysis of NoSQL datastores as well as a traditional relational database system in terms of their geospatial querying capabilities.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; **Geographic information systems**; *Key-value stores*; • **General and reference** → *Surveys and overviews*; • **Software and its engineering**;

KEYWORDS

Data stores, Geospatial functionality, Spatio-temporal characteristics, Spatio-temporal databases

ACM Reference Format:
Antonios Makris, Konstantinos Tserpes, Dimosthenis Anagnostopoulos, Mara Nikolaidou, and Jose Antônio Fernandes de Macedo. 2019. Database system comparison based on spatiotemporal functionality. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3331076.3331101

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331101

1 INTRODUCTION

Nowadays, massive volumes of spatiotemporal data are constantly being generated by many scientific, engineering and business applications. For example, remotely sensed data from NASA's Earth Observing System produces 1 TB of data each day [1] while the Automatic Dependent Surveillance Broadcast (ADS-B) system which gathers information about position, identification and course of aircrafts, produces 285 billion points per year [2]. Also, the Department of Computing in Federal University of Ceará (UFC) tracks vehicle movements in the area of Fortaleza generating huge volumes of data (Figure 1).

Spatiotemporal database management systems (STDBMSs) constitute core components in tackling the challenges of spatio-temporal applications. They allow for the efficient management of the data and the application of complex queries.

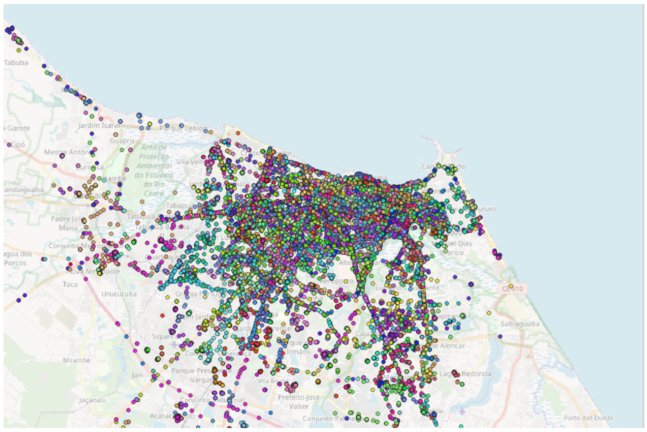


Figure 1: A small fraction of data points from vehicles moving in Fortaleza area, gathered from UFC.

As expected, the demand for high quality of service and increased performance introduces new non-functional requirements that these systems need to cope with. Such requirements include:

scalability, availability, reliability, consistency, performance and accuracy.

Towards this end, distributed computing is considered to be an enabling technology. Popular implementations of DBMS are based on distributed computing architectures and the majority of them are providing for the managements of spatio-temporal data in their core functionality (e.g. Redis[1], HBASE [3], MongoDB[2] and PostgreSQL [4]).

The goal of this paper is to highlight these spatio-temporal functionalities of DBMSs and present the key architectural characteristics that are supporting them. It goes on to perform a comparison between a relational database system and several widely used NoSQL datastores across those characteristics. Based on the literature review and to the best of our knowledge the most prominent differences concerning geospatial support for these data stores, are their data model and a number of geospatial capabilities. Such capabilities relate to geospatial indexing used, the geometry types supported and the spatial query operators performed.

The rest of this paper is organized as follows. Section 2 presents a categorization of the spatio-temporal data types. Section 3 presents the key-characteristics of the database systems examined. Section 4 highlights the geospatial capabilities while Section 5 presents the final conclusions.

2 SPATIO-TEMPORAL DATA TYPES

The spatio-temporal data types can be divided into two major categories: spatial data and temporal data as shown in Figure 2.

Within the spatial referenced data group, the data can be further classified into two different types, raster and vector. Generally, point, line, and polygon are primitive data types of vector type. Point data have zero dimensions and are used to represent non-adjacent features and discrete data points. Line data are used to represent linear features. Line features have a starting and ending point and the one dimension of which they are composed can be used to measure length. Polygons are used to represent areas such as the boundary of a city. Polygon features are two dimensional and therefore can be used to measure the area and perimeter of a geographic feature. These spatial abstract types have several common properties such as coordinates within a reference system and operations like calculation of distance or containment. On the other hand, raster data types (grid data) are cell-based and represents surfaces, aerial and satellite imagery.

Concerning temporal data, spatio-temporal databases support three kind of time. Transaction time which is the time that an object was presented as stored record in the database. The temporal aspects of an element evolve discretely and this kind of time is used to trace past states of objects. Valid time which is the time that an object has existed in reality. The temporal aspect of an element change continuously and this kind of time is applied to facts and events and used on object attributes and relationships between objects. Bitemporal time is used to trace the evolution of a dynamic collection of valid time facts and is a combination of transaction and valid time. The valid time can be of the event or period type while the transaction time can be of the interval type.

3 DATABASE SYSTEMS

This section presents the key-characteristics of the database systems compared. Redis, MongoDB, Neo4j and HBase belong to a wider category called NoSQL which describes a large class of DBMS. These systems do not follow the rules of the traditional relational DBMS and also do not use the traditional SQL queries over the data. NoSQL-based systems are often open source projects and are designed to process and handle very large datasets which are particularly prone to performance problems caused by the limitations of SQL and the relational model of databases. These systems typically run on cluster computers made from commodity hardware, provide "shared nothing" horizontal scalability, can support a large number of concurrent users and deliver highly responsive experiences to a globally distributed base of them. Subsequently, they provide dynamic schema and can handle semi- and un-structured data. Based on the data model, the NoSQL data stores can be classified into four major types: *key-value stores*, *column-family stores*, *document stores*, and *graph stores*. We consider a representative and widely used system from each type that include spatial extensions and provide a license-free installation.

One representative *key-value store* is Redis. Specifically, Redis is an in-memory key-value store, used as a database, cache and message broker. It supports various data structures including Strings, Lists, Sets, Sorted Sets, Hashes, Bitmaps and HyperLogLogs making it extremely powerful and allowing the execution of complex client functionality. As mentioned, Redis is an in-memory system, means that operations are executed extremely efficiently in memory and for this reason different functionalities can be achieved with low complexity, least amount of overhead on the network and low latency. Thus, it can handle extremely high throughput (millions of operations per second) compared with other partially disk-based database solutions that requires a large cluster of nodes to handle high-volume real time updates.

The distributed implementation of Redis is called Redis Cluster. The nodes in Redis Cluster are responsible for holding the data, capturing the state of the cluster and mapping keys to the right nodes. The nodes in the cluster are connected through a service channel using a TCP bus and a binary protocol, called the Redis Cluster Bus [5]. To exchange and propagate information about the cluster, nodes use a gossip protocol in order to auto-discover new or existing nodes, to send ping-pong packets, to detect working or non-working nodes, to send cluster messages, to trigger specific conditions and to promote slave nodes to master when needed in order to continue to operate when a failure occurs [6].

Apache HBase is an open source distributed *column-family based datastore* built on top of HDFS that provides high scalability and fault-tolerance. Data are stored in labeled versioned tables which in turn stored as multidimensional sparse maps. Each table version represents an auto-assign timestamp created at cell creation time and on table creation a set of column families are defined. Each table consist of rows and columns where each row contains a sorting key and an arbitrary number of columns. Every column can have several versions for the same row key. Every cell is tagged by a column name and family while each row is sorted by a row key which serves as primary key. Select queries are executed based on table's primary key and each scan results into a MapReduce job [3].

[1] Redis, https://redis.io/

[2] MongoDB, https://www.mongodb.com/

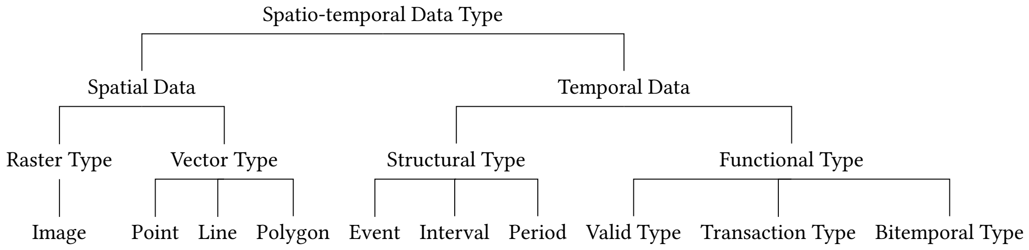


Figure 2: Spatio-temporal data types

Each MapReduce job consist of a master node which is responsible for holding the cluster state, for assigning regions to regionservers and for recovering regionserver in case of failure [7]. It is worth noting that HBase clones BigTable from Google. The data model of these two system are very similar. Tables in HBase are automatically partitioned horizontally into regions. Every region contains a subset of table's rows. Similar to HDFS and MapReduce, HBase supports master/slave architecture. The three major components of the HBase are: HBaseMaster which has the responsibility for assigning regions to HRegionServes, HRegionServers which has the responsibility to handle client read and write requests and HBase client which has the responsibility to find HRegionServers that are serving a particular row range [8].

MongoDB is an open source *document based datastore* which supported commercial by 10gen. Although MongoDB is non-relational, it implements many features of relational databases, such as sorting, secondary indexing, range queries and nested document querying. The use of operators like create, insert, read, update, remove as well as manual indexing, indexing on embedded documents and index location-based data also supported. In such systems, data are stored in collections called documents, consist of entities that provide some structure and encoding of the managed data. A collection is similar to a table in relational databases and is schema-free, which means that documents with different data structures can be stored in same collection [9]. Each document constitutes an associative array of scalar value, lists or nested arrays and has a unique special key "_id" which is a 24-bits string calculated by the timestamp, host identifier, process identifier (PID) and a counter and used for explicitly identification. This field and the actual document are conceptually similar to a key-value pair. In each _id field is created a unique index by default [10].

MongoDB documents are serialized naturally as Javascript Object Notation (JSON) objects and stored internally using a binary encoding of JSON called BSON. Each BSON's maximum size is limited to 16MB. As all NoSQL systems, in MongoDB there are no schema restrictions and can support semi-structure data and multi-attribute lookups on records which may have different kinds of key value pairs [11]. In general, documents are semi-structured files like XML, JSON, YALM and CSV. For storing data in MongoDB there are two ways: a) nesting documents inside each other, an option that can work for one-to-one or one-to-many relationships and b) reference to documents than nesting the entire document, an option that the referenced document only retrieved when the user requests data inside this document [12].

Neo4j is a *graph based datastore*. It doesn't provide a standard SQL interface but direct REST requests. In these systems, a graphical representation is used which can address scalability concerns. Graph structures are composed of edges, nodes and properties which provides index-free adjacency. Nodes and edges consist of objects with embedded key value pairs. Graph databases are specialized on efficient management of heavily linked data and are optimized for highly connected data. In such systems cost intensive operations like recursive joins can be replaced by efficient graph traversal and graph pattern matching techniques. In case of graph traversal, the query processing starts from one node and then the other nodes are traversed based on the description query while on graph pattern matching techniques the defined pattern located in the original graph. Neo4j contains a mini-index in each vertex and edge of the objects connected to the graph and this typically means that the size of the graph has no performance impact upon a traversal as well as the cost of a local step (hop) remains the same. Also a global adjacency index is used to locate the starting point of a traversal. Indexes provide a fast and efficient way to retrieve vertices based on their values [13].

On the other hand, PostgreSQL belongs to the category of traditional relational database management systems (RDBMS) and it is widely adopted in industrial and research settings. PostgreSQL is an open source object relational database system (ORDBMS) that uses and extends the SQL language [14]. It allows several well-known operations such as inserts, updates, deletes etc. and queries in data that stored in database [4]. A fundamental characteristic of PostgreSQL database is the support of user-defined objects including data types, functions, operators, domains and indexes. It supports multiple operators for querying, filtering, joining, grouping and modifying data.

Some characteristics of such systems that have a significant impact on the scalability of the data stores are the data model, query model, partitioning, consistency and replication. The query model refers to data retrieval commands and querying languages that used to retrieve data that stored in the database. A commonly employed strategy for storing and processing massive datasets is the partition of the data across different server nodes, thus achieving high availability and fault tolerance. Replication relates to dependability on database systems and refers to the process by which the same data are stored on multiple servers so that read and write operations can be distributed over them. Replication also provides fault tolerance because data availability can withstand the failure of one or more servers. Strongly related to replication is the consistency

Table 1: Summary of key characteristics

Database System	Data Model	Query Model	Partitioning	Consistency	Replication
Redis	Key-value store	Data retrieval commands with no queries or query planner abstractions in the middle	Range partitioning, Hash partitioning, Consistent hashing	Eventual consistency	Master-slave asynchronous replication
HBase	Column family store	Shell like command query. REST and Thrift API are supported	Tables are partitioned by row-key into regions stored in different region server	Strong consistency as each record must be updated on assigned region server and replication committed before read	HDFS to store replication with selectable factors
MongoDB	Document store	Queries as BSON objects sent to MongoDB driver	Range partitioning based on a shard key	Immediate consistency	Master-slave asynchronous replication
Neo4j	Graph store	Cypher query language match patterns of nodes and relationships in the graph	Cache-based	Eventual consistency	Master-slave
PostgreSQL	ORDBMS	Utilizes the SQL querying language	Range partitioning, List partitioning	Eventual consistency-asynchronous write, Strong consistency-synchronous write (Serializable Transactions, Explicit Blocking Locks)	Streaming replication, Synchronous replication

level provided by the data store. Consistency is a system property that ensures that a transaction brings the database from one valid state to another. The consistency models are: strong, eventual or immediate consistency. Strong or immediate consistency ensures that when write requests are confirmed, the same (updated) data are visible to all subsequent read requests. In eventual consistency, changes eventually propagate through the system given sufficient time and therefore some server nodes may contain outdated data for a period of time. In general, in distributed systems there is a trade-off between consistency and availability of data. Table 1 presents the examined systems classified based on the criteria of data model, query model, partitioning, consistency and replication. These criteria have a significant impact on the scalability of the data stores.

4 SPATIO-TEMPORAL FUNCTIONALITY

In this section are presented the spatio-temporal functionalities and the geospatial capabilities between the examined database systems. The most prominent differences concerning geospatial support relate to indexing, geometry types and query operators.

Redis can efficiently handle and support geospatial data with the use of geospatial set and operations that can handle location-specific indexing, searching, updating and sorting in a simple way. With the combination of the built-in functions and data types, the infrastructure may provide reduced code complexity, reduced network bandwidth consumption and overall faster execution [15].

For geospatial indexing, Redis uses the *Geo Set*. Geo Set is a data structure, implemented similar to another data structure called Sorted Set (basic data structure of Redis) and it is the basis for working with geospatial data. Each Geo Set include a unique identifier and a coordinate pair (longitude, latitude). Several functions that used for geospatial index management are: Creation, Adding, Updating, Removing, Deleting, Reading and Searching the index with a list of geospatial commands (GEOADD, GEOPOS, GEOHASH, GEODIST etc.). Redis Geo Set allows storing and querying various geometry types such as: Point, Polygon, MultiPolygon, MultiPoint, LineString, MultiLineString and GeometryCollection.

As mentioned above, the Geo Set data structure is similar to a Sorted Set. A Sorted Set is a mix between a Set and a Hash. Like Sets, it contains unique string elements and every element is associated with a floating point value, called score, just like Hash. The elements inside a Sorted Set sorted by their score allowing ordering and searching for members by their rank or score, which is 64-bit floating point number. The main difference between a Sorted Set and a Geo Set is that the score in the latter is used for store the location. Location in substance constitutes a coordinate pair, longitude and latitude. The main functionality Geo Set provides, is the encoding and decoding of such coordinate pairs and numerical scores. For translating these two representations Redis implements the Geohash system.

Geohash algorithm is a latitude/longitude geocode system that used for encoding and decoding coordinate pairs in a compact form and divides geographic regions into a hierarchical structure [16]. Coordinates are converted into a string using a base-32 character map. A Geohash string represents a spatial bounding box, thus Geohash divides geographic space into buckets of grid shape [17]. Redis uses Geohash to map coordinate pairs and their hash values and stores in each member's score the hash's numerical representation. The Geohash system divides the world into rectangular cells where each such cell is uniquely identified by its hash value. The cells' hashes are computed by interleaving the information from both location coordinates into a single value. At first, the algorithm takes a coordinate and tests if the longitude is on the left or right of the Prime Meridian. If it's in the right then the hash's most significant bit is turned on to 1 otherwise is turned to 0. In the next step the same logic applied to latitude. If lies north to the Equator the hash's next bit is turned to 1 leading to the binary value 11. This process continues by dividing that hash's cell according to the longitude again and by repeating Geohash algorithm, the resulting binary value represents a location with increasing degree of accuracy. The accuracy is proportional to the number of iterations that are performed. Redis allows 26 Geohash iterations at max that produce 52-bit long hash values which in turn provide an accuracy error at about 0.6 m. Figure 3 illustrates the Geohash algorithm with the logic behind bits representation [15].

Geospatial indexing through Geohash, only supports two spatial dimensions with no regard to altitude. To manage 3-dimensional geospatial data, Redis uses a combination of two well-known data structures, GeoSet that used for storing coordinates pairs and SortedSet for storing the elevation of each member's score through a xyzset module.

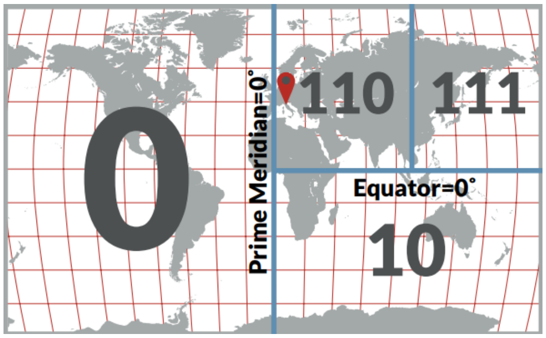


Figure 3: Geohash algorithm

HBase fails to provide in-built spatio-temporal querying capability. For spatial functionality, a scalable data storage solution called HBaseSpatial exists as shown in [18]. In general, the system provides efficient and effective distributed storage and vector data indexing and can support large scale spatial vector data storage and management. HBaseSpatial is divided into two main parts, the storage model and the index model. The storage model receives the vector data from shapefiles then puts these data into the index model and finally converts these data to WKB type and stores them in the HBase table. The index algorithm calculates the id of each vector data and put them into the index table. Range searches are efficiently improved by the use of this secondary index method. Vector spatial data include attribute and topology data and spatial coordinates. Spatial attributes of such data contain a large number of geometry coordinates and for this reason the WKB format is used for storing the information in binary format. For indexing, the grid spatial partition index method is used, where the global scope of the latitude and longitude pair is divided into different levels of the grid.

An another system that provides spatial functionality to HBase is presented in [19]. STEHIX (Spatio-TEmporal Hbase IndeX) index structure is suitable to process spatio-temporal queries and it is a two-level lookup mechanism of HBase. At first, with the use of Hilbert curve, geolocation data linearized and stored in a meta table and then for each region an index mechanism is used for storage files. Also a system called MD-HBase [20] adds an index structure to meta table of HBase but does not provide an index to lookup inner data of regions. Finally, a system called GeoMesa[3] provides spatio-temporal indexing on top of many systems including HBase. GeoMesa is an open source tool that enables large scale geospatial querying and analytics on distributed computing systems.

For spatial functionality in MongoDB, data are stored either as GeoJSON objects which is a format for encoding a variety of geographical data structures or as legacy coordinate pairs (MongoDB versions 2.2 and earlier). GeoJSON supports a) Geometry types as: Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon and GeometryCollection b) Feature, which is a geometric object with additional properties and c) FeatureCollection, which consist a set of features [21]. Each GeoJSON document is composed of two fields: i) Type - the shape being represented, which informs a GeoJSON reader how to interpret the "coordinates" field and b) Coordinates - an array of points, the particular arrangement of which is determined by "type" field. In MongoDB, the geographical representation need to follow the GeoJSON format structure in order to be able to set a geospatial index on the geographic information. MongoDB supports BTree indexes (not R-trees) to support specific types of data and queries such as: Single Field, Compound Index, Multikey Index, Text Indexes, Hashed Indexes and Geospatial Index. To support efficient queries on geospatial coordinate data, MongoDB provides two special indexes: 2d indexes that uses planar geometry when returning results and 2dsphere indexes that use spherical geometry to return results. A 2dsphere index supports queries that calculate geometries on an earth-like sphere and supports all MongoDB geospatial queries: queries for inclusion, intersection and proximity. 2d indexes support queries that calculate geometries on a two-dimensional plane i.e. queries that interpret geometries on a flat surface and some spherical queries but do not support GeoJSON-formatted queries or GeoJSON data values. Also MongoDB supports Geo Haystack index that used to query small areas but nowadays it is less used by applications. MongoDB computes the geohash values for the coordinate pairs and then indexes the geohash values. Concerning spatio-temporal functionality MongoDB supports four geospatial query operators: $geoIntersects, $geoWithin, $near and $nearSphere.

In Neo4j, a plugin called Neo4j-Spatial exists and supports various geometry types such as Geometry, Point, LineString, Polygon, MultiPoint, MultiLinestring and MultiPolygon. The spatial queries

[3] GeoMesa, https://github.com/locationtech/geomesa

Table 2: Summary of spatio-temporal key characteristics

Database System	Geometry Types	Geospatial Indexing	Spatial query operators-functions
Redis	Point, LineString, Polygon, MultiPoint, MultiLineString MultiPolygon, GeometryCollection	Geo Set	Geoadd, Geopos, Geohash, Geodist, Geopathlen, Georadius, Georadiusbymember, Geoencode, Geodecode, Geometry filter
HBaseSpatial	Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, SimpleFeatureType, GeometryCollection	Grid spatial index method	Range queries of vector spatial data, k-NN queries
MongoDB	Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, GeometryCollection, Feature (geometric object with additional properties), FeatureCollection (a set of features)	2dsphere, 2d	$geoIntersects, $geoWithin, $near, $nearSphere
Neo4j	Geometry, Point, LineString, Polygon, MultiPoint, MultiLinestring, MultiPolygon	RTree	Contain, Cover, Covered By, Cross, Disjoint, Intersect, Intersect Window, Overlap Touch, Within, Within Distance, Area, BBox, Boundary, Distance, Buffer, Centroid, ConvexHull, Envelope
PostgreSQL	Point, LineString, Polygon, MultiPoint, MultiLineString, MultipPolygon, GeometryCollection	Generalized Search Tree (GiST)	ST_Within, ST_Intersects, ST_DWithin + Order by distance, ST_Area ...

implemented, include the following topological functions: Contain, Cover, Covered By, Cross, Disjoint, Intersect, Intersect Window, Overlap, Touch, Within and Within Distance. Moreover the analysis functions provided are: Area, BBox, Boundary, Distance, Buffer, Centroid, ConvexHull, and Envelope and the set functions include Difference,Intersection, Union and SymDifference methods[4]. Neo4j-Spatial can import data in both ESRI Shapefile (SHP) and Open Street Map (OSM) formats. Each format provides different layers which in turn support different geometry types. A single layer can be divided into multiple sub-layers through the use of pre-configured filters, which can be proven efficient when working with large datasets. In addition, each spatial data layer has its own configuration of the coordinate system obtained from the input files (SHP or OSM) [22]. Concerning indexing, Neo4j-Spatial uses RTree for spatial queries which is suitable for 2-dimensional and 3-dimensional spatial data. Typically, with the use of an RTree, every geometry is grouped and represented with its minimum bounding rectangle in the next-higher level of the tree. For graph traversal, where a node or a relationship needs to be found based on a property, a spatial lookup is performed for an increased performance. The system uses an R-tree index structure only to retrieve the start elements and from that point onwards an index-free traversal is executed through the graph [23].

In PostgreSQL, there is a special extension called PostGIS that integrates several geofunctions and support geographic objects. PostGIS contains more than one thousand geofunctions [21] and according to [4] can be divided into five categories: management, conversion, retrieval, comparison and generation. In general, PostGIS provide spatial services such as spatial objects, spatial indexes, spatial operators and spatial manipulation functions [24]. It supports several geometry types as: Points, LineStrings, Polygons, MultiPoints, MultiLineStrings, MultipPolygons and GeometryCollections. In general PostGIS implementation is based on "light-weight" geometries and the indexes are optimized to reduce disk and memory footprint.

Concerning indexing, PostgreSQL supports several types of indexes such as BTree, RTree, Hash, Generalized Inverted Indexes (GIN) and Generalized Search Tree (GiST) called R-tree-over-GiST. BTree is the default type of index used in one-dimensional ordered data and can be used efficiently for equality and range queries with all datatypes. With the use of RTree indexing, data are divided into rectangles and this index is suitable for two-dimensional spatial data. For general balanced tree structures and high-speed spatial querying, PostgreSQL uses GiST indexes that can be used to index the geometric data types. GiST stands for "Generalized Search Tree" and is suitable for speeding up search queries on all kinds of irregular data structures. BTree on the other hand cannot accomplish this functionality and GiST have two advantages over RTree and Btree. At first, GiST can used to index columns with null values and moreover can support the concept of "lossiness" as mention in [25], which means that in case of large GIS object, only the significant part of an object is stored, just the bounding box. Moreover, GIS objects larger than 8K will lead RTree indexes in failure.

Table 2 presents the examined systems classified based on the spatio-temporal criteria of geometry types, geospatial indexing and spatial query operators.

5 CONCLUSIONS

As more applications are dependent on data of spatio-temporal nature, the DBMS community will continue to seek more efficient

[4] Neo4j Spatial, https://github.com/neo4j-contrib/spatial

ways to support them. Redis, HBase, MongoDB, Neo4j and PostgreSQL, are all representative cases of DBMS that provide geospatial querying capabilities. Each of those systems is based on a different underlying technology and model resulting in varying geospatial indexing methods, geometry types and spatial query operators.

ACKNOWLEDGMENTS

The research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under the HFRI PhD Fellowship grant (GA. no. 2158).

This work has been developed in the frame of the MASTER project, which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 777695.

REFERENCES

[1] G Leptoukh. Nasa remote sensing data in earth sciences: Processing, archiving, distribution, applications at the ges disc. In *Proc. of the 31st Intl Symposium of Remote Sensing of Environment*, 2005.

[2] Automatic dependent surveillance-broadcast (ads-b). https://www.faa.gov/nextgen/programs/adsb/. Accessed: 2018-11-14.

[3] Mehul Nalin Vora. Hadoop-hbase for large-scale data. In *Computer science and network technology (ICCSNT), 2011 international conference on*, volume 1, pages 601–605. IEEE, 2011.

[4] Sarthak Agarwal and KS Rajan. Performance analysis of mongodb versus postgis/postgresql databases for line intersection and point containment spatial queries. *Spatial Information Research*, 24(6):671–677, 2016.

[5] Antonios Makris, Konstantinos Tserpes, and Dimosthenis Anagnostopoulos. Load balancing in in-memory key-value stores for response time minimization. In *International Conference on the Economics of Grids, Clouds, Systems, and Services*, pages 62–73. Springer, 2016.

[6] Antonios Makris, Konstantinos Tserpes, and Dimosthenis Anagnostopoulos. A novel object placement protocol for minimizing the average response time of get operations in distributed key-value stores. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 3196–3205. IEEE, 2017.

[7] Dorin Carstoiu, Elena Lepadatu, and Mihai Gaspar. Hbase-non sql database, performances evaluation. In *in Computer Science (1986), Master in Computer Science (1990), and PhD in Computer Science*. Citeseer, 2010.

[8] Ankur Khetrapal and Vinay Ganesh. Hbase and hypertable for large scale distributed storage systems. *Dept. of Computer Science, Purdue University*, pages 22–28, 2006.

[9] Xiaomin Zhang, Wei Song, and Liming Liu. An implementation approach to store gis spatial data on nosql database. In *Geoinformatics (GeoInformatics), 2014 22nd International Conference on*, pages 1–5. IEEE, 2014.

[10] Veronika Abramova and Jorge Bernardino. Nosql databases: Mongodb vs cassandra. In *Proceedings of the international C* conference on computer science and software engineering*, pages 14–22. ACM, 2013.

[11] Antonios Makris, Konstantinos Tserpes, Vassiliki Andronikou, and Dimosthenis Anagnostopoulos. A classification of nosql data stores based on key design characteristics. *Procedia Computer Science*, 97:94–103, 2016.

[12] Zachary Parker, Scott Poe, and Susan V Vrbsky. Comparing nosql mongodb to an sql db. In *Proceedings of the 51st ACM Southeast Conference*, page 5. ACM, 2013.

[13] Justin J Miller. Graph database applications and concepts with neo4j. In *Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA*, volume 2324, page 36, 2013.

[14] Antonios Makris, Konstantinos Tserpes, Giannis Spiliopoulos, and Dimosthenis Anagnostopoulos. Performance evaluation of mongodb and postgresql for spatiotemporal data. 2019.

[15] Redis for geospatial data. https://redislabs.com/docs/redis-for-geospatial-data/. Accessed: 2018-7-15.

[16] Zoran Balkić, Damir Šoštarić, and Goran Horvat. Geohash and uuid identifier for multi-agent systems. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 290–298. Springer, 2012.

[17] Jiajun Liu, Haoran Li, Yong Gao, Hao Yu, and Dan Jiang. A geohash-based index for spatial data management in distributed memory. In *Geoinformatics (GeoInformatics), 2014 22nd International Conference on*, pages 1–4. IEEE, 2014.

[18] Ningyu Zhang, Guozhou Zheng, Huajun Chen, Jiaoyan Chen, and Xi Chen. Hbasespatial: A scalable spatial data storage based on hbase. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 644–651. IEEE, 2014.

[19] Xiaoying Chen, Chong Zhang, Bin Ge, and Weidong Xiao. Spatio-temporal queries in hbase. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 1929–1937. IEEE, 2015.

[20] Shoji Nishimura, Sudipto Das, Divyakant Agrawal, and Amr El Abbadi. Md-hbase: A scalable multi-dimensional data infrastructure for location aware services. In *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, volume 1, pages 7–16. IEEE, 2011.

[21] Stephan Schmid, Eszter Galicz, and Wolfgang Reinhardt. Performance investigation of selected sql and nosql databases. *AGILE 2015–Lisbon*, pages 9–12, 2015.

[22] Elaheh Pourabbas. *Geographical information systems: trends and technologies*. CRC press, 2014.

[23] Chad Vicknair, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, and Dawn Wilkins. A comparison of a graph database and a relational database: a data provenance perspective. In *Proceedings of the 48th annual Southeast regional conference*, page 42. ACM, 2010.

[24] Lijing Zhang and Jing Yi. Management methods of spatial data based on postgis. In *Circuits, Communications and System (PACCS), 2010 Second Pacific-Asia Conference on*, volume 1, pages 410–413. IEEE, 2010.

[25] Paul Ramsey et al. Postgis manual. *Refractions Research Inc*, 2005.

A Spatial Index for Hybrid Storage

Athanasios Fevgas
Data Structuring & Engineering Lab
ECE Department, University of Thessaly
Volos, Greece
fevgas@e-ce.uth.gr

Panayiotis Bozanis
Data Structuring & Engineering Lab
ECE Department, University of Thessaly
Volos, Greece
pbozanis@e-ce.uth.gr

ABSTRACT

The introduction of flash SSDs has accelerated the performance of DBMSes. However, the intrinsic characteristics of flash motivated many researchers to investigate new efficient data structures. The emergence of 3DXPoint, a new non-volatile memory, sets new challenges: 3DXPoint features low latency and high IOPS even at small queue depths. However, the cost of 3DXPoint is 4 times higher than that of a flash-based device, rendering hybrid storage systems a good alternative. In this paper we pursue exploiting the efficiency of both 3DXPoint and flash-based devices introducing H-Grid, a variant of Grid-File for hybrid storage. H-Grid uses a flash SSD as main store and a small 3DXPoint device to persist the hottest data. The performance of the proposed index is experimentally evaluated, comparing it against GFFM, a flash efficient implementation of Grid File. The results show that H-Grid is faster than GFFM execution on a flash SSD, reducing the single point search time from 35% up to 43%.

CCS CONCEPTS

• **Information systems → Data access methods**.

KEYWORDS

Spatial Index, Grid File, SSD, 3DXPoint, Optane

ACM Reference Format:
Athanasios Fevgas and Panayiotis Bozanis. 2019. A Spatial Index for Hybrid Storage. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3331076.3331091

1 INTRODUCTION

The emergence of non-volatile memories (NVM) has enabled new storage devices with amazing features like ultra-high read and write speeds, small size, low power consumption and shock resistance. Nowadays, flash-based solid state drives (SSDs) are found in the vast majority of consumer computer systems, as well as in almost every data center. Data intensive applications, like DBMSes, have drawn significant performance advantages by this evolution. As a result, index structures for flash SSDs have become a promising field of study for many researchers. Most of the presented works concern tree indexes for one- [1, 11, 16, 25, 26] and multi-dimensional data [3, 10, 14, 28, 30], while fewer exist investigating flash efficient hashing methods [5, 17]. Briefly, the majority of proposals aim at reducing the number of small random writes that deteriorate the performance of SSDs while avoiding the mingling of reads and writes for the same reason. On the other hand, they seek to exploit the high internal parallelism of modern devices. Thus, some well known techniques which are employed to meet these objectives are: i) postponing of write operations and performing them in batches, ii) buffering of retrieved read pages, iii) applying logging and iv) grouping of page read requests.

A new class of SSDs was introduced by Intel, under brand name "Optane", earlier in 2017. These storage devices are based on 3DXPoint non-volatile memory technology. 3DXPoint uses a layered crosspoint architecture, permitting individual addressing of each memory cell. Opposite to flash, it supports in-place-writes, relieving the SSD controller from the burden of maintaining out-of-place updates and garbage collection operations. It provides up to 10^3 better access times compared to NAND flash while its density is 10 times higher than that of DRAM [22]. Therefore, [8] proposed two more possible uses of 3DXPoint, other than as secondary storage: i) as a low cost extension of DRAM, and ii) as persistent main memory directly accessed by the CPU.

The efficiency of a storage device is described by three performance metrics: IOPS, bandwidth and latency. IOPS determine the number of I/O operations that the device is able to carry out over the unit of time. On the other hand, the bandwidth expresses the throughput that a drive can deliver, measured in MBs/sec. Finally, latency is the amount of time that an I/O request takes to complete, i.e., the response time of an operation. Latency is of paramount importance for the efficiency of a storage system, since low latency is tightly connected with better user experience. Little's law [18] for storage systems mandates that $IOPS = \frac{Queue}{Latency}$, where Queue is the number of outstanding requests, i.e. the number of I/O requests sent to the device in parallel. It is clear that reducing latency retains IOPS efficiency even with less concurrent I/O. Lower latency values enable workloads to finish into a fraction of the initial time. According to [32] the latency of high-performance NVMe SSDs contributes over 19% of the overall response time on online applications. New SSD devices have been introduced lately providing ultra-low latency; such devices are Intel's Optane series and Samsung's Z-NAND. Intel Optane SSDs are based on 3DXPoint non-volatile memory and can provide a latency reduction of one order of magnitude compared to the conventional NAND flash SSDs. 3DXPoint SSDs can deliver high IOPS even when a small number of concurrent outstanding I/O is used (small queue depth), while

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331091

their NAND counterparts are more efficient under large batched I/O [8, 13].

Previous works for flash efficient database indexes focus on exploiting the high internal parallelism of SSD devices by issuing multiple read or write operations at once. Several works utilize large queue depths, pursuing to distribute the workload among multiple NAND chips, accelerating query performance [27]. Although this technique has been proved very useful so far, the low latency of new NVM technologies can further improve the performance, especially where limited opportunity for grouping I/O requests exists. To the best of our knowledge, this is the first time that the low latency 3DXPoint NMV is exploited to accelerate the performance of a spatial index.

The contributions of this paper can be summarized as follows:

- We introduce a new spatial index structure, the H-Grid (Hybrid Grid-File), which is designed for hybrid storage. Particularly, we consider flash SSDs as the mass storage tier and 3DXPoint ones as the performance tier.
- We present a hot region detection algorithm that recognizes regions of high interest, storing them to the performance tier.
- We evaluate our H-Grid through extensive experimentation, utilizing one real and two synthetic datasets. We study single point search queries, region and kNN queries as well.

The remainder of this paper is organized as following. Section 2 describes the related work in hybrid storage systems and hybrid indexes. The design and implementation details of H-Grid are unfolded in Section 3. Section 4 presents the experimental results and, finally, our conclusions are listed in Section 5.

2 RELATED WORK

Hybrid storages are not rare in database systems; several algorithms have been proposed so far [24]. Most of the related works, until now, consider flash-based solid state drives as the performance tier and magnetic disks as the storage tier. In fact, hybrid storage systems employ SSDs either as a cache between main memory and HDD or as high performing devices storing permanently the hottest data.

In [2] a flash based SSD acts as an extension of the standard main memory bufferpool accommodating high priority data. The hot data regions are identified using frequency and recency statistics, while an aging mechanism ensures that the cached regions are in line with the I/O pattern, as it changes over the time. The authors in [19] study different buffer management policies in relational DBMSes (i.e. MySQL), when a hybrid SSD/HDD scheme is used as persistent storage. Their findings indicate that the performance of hybrid systems, which employ SSDs for caching, is highly dependent on the ratio between SSD and HDD bandwidth.

Hystor [4] is an extension to Linux operating system that identifies hot and performance critical data blocks by monitoring the I/O sequence. This data is stored on a fast SSD instead of a magnetic disk. Following a different roadmap, MOLAR [20] proposes the implementation of the hot page detection mechanism into the SSD's controller. Simulated experiments have shown that MOLAR can reduce the average write latency in SSDs by 3.5 times.

The efficiency of hybrid storage systems is connected with the accuracy of hot data identification. [21] uses a probabilistic algorithm to locate hot data. The algorithm maintains two probabilities. The first probability contributes to the decision of which pages should be evicted from RAM and the second one determines the persistent storage (SSD/HDD) an evicted page should be moved to.

Although many flash efficient database indexes have been proposed so far, there exist only a few hybrid ones. The *HybridB tree* [12] is a B+tree variant for hybrid SSD/HDD storage. It always keeps the internal nodes in the SSD, while it distributes the leaf node pages between HDD and SSD. Specifically, it adopts a huge-leaf organization for the leaf nodes, aiming to reduce costly splits and merges. A huge-leaf occupies two or more pages in the secondary storage and includes a special node for metadata and a logging part as well. The XB+Tree [15] is a hybrid index for PCM/RAM memory. PCM is utilized as non-volatile random access memory along with DRAM rather than as a secondary storage. The proposed index distinguishes nodes according to their read/write tendency, retaining write intensive nodes in DRAM, while it stores the read intensive ones to PCM. In this way it reduces costly write operations in PCM, while simultaneously increases the overall performance.

A recent study [31] investigates the use of 3DXPoint technology to enhance the performance of database systems. Specifically, the authors recognize write amplification, careless use of temporary tables and bufferpool cache misses as factors that degrade query performance. In the sequel, they experimentally show that an enterprise class 3DXPoint SSD can improve query performance by 1.1-6.5x compared to a flash counterpart.

3 THE H-GRID

Spatial data structures are of paramount importance for spatial query processing. They represent simple or complex spatial objects (e.g. points, lines polygons, etc) in a manner that simplifies execution of spatial queries [29]. In our previous work [5, 6] we utilized flash SSDs to enhance the efficiency of Grid File [23]. In this paper, our objective is to take advantage of a new non-volatile memory technology, the 3DXPoint. Therefore, we introduce the H-Grid, a Grid File variant for hybrid storage.

3.1 H-Grid Design

A common method on past research for flash efficient database indexes is to group I/O operations, exploiting the high bandwidth, the internal parallelism of modern SSDs and the efficiency of NVMe protocol. This strategy provides sufficient results, especially in range and kNN queries as they usually involve access to multiple pages. Furthermore, some authors propose grouping of incoming single key search requests into sets that are processed simultaneously [27]. However, in all aforementioned cases, accessing the upper level nodes of tree-indexes does not always exploit the full bandwidth of SSDs, even if multiple nodes are fetched with a single I/O request. The performance characteristics of 3DXPoint, i.e. its efficiency even at small size I/O, motivated us to introduce H-Grid. H-Grid is a Grid-File variant designed for hybrid, 3DXPoint/flash storage. It exploits a frequency based model for data placement. It detects performance critical regions placing them to the low latency 3DXPoint storage, while it leaves the rest of them to the flash SSD. To the best

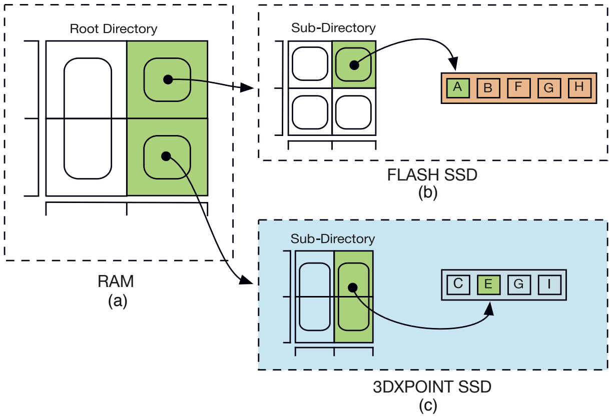


Figure 1: Overview of H-Grid. A part of the Grid-File is migrated to the 3DXpoint storage.

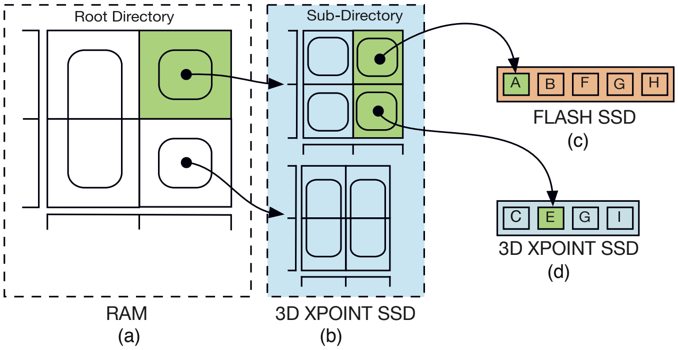


Figure 2: H-Grid special case. All sub-directories are hosted to the 3DXPoint SSD.

of our knowledge, H-Grid is the first attempt to introduce a spatial index that exploits hybrid 3DXPoint/flash I/O.

A running example of H-Grid is illustrated in Figure 1. The H-Grid implementation follows the two-level Grid File design as it is presented in [9]. Thus, H-Grid employs a small, memory resident Root Directory (Fig. 1a) and many sub-directories that reside in the physical storage. The sub-directories hold the addresses of data buckets that contain the actual data. The sub-directories and the data buckets can reside in either a flash (Fig. 1b), or a 3DXPoint SSD (Fig. 1c). A selection algorithm locates frequently accessed regions that are eligible for the 3DXPoint storage, considering weight values for each retrieved directory or data bucket page. These weights are calculated using the access frequencies of the pages. We use two hashing tables (one for directory and one for data pages) to associate each page in the 3DXPoint storage with its corresponding weight value.

H-Grid leverages in-memory buffers to accommodate pages that are either retrieved from the SSDs or temporary stored prior to a batch write operation [5, 6]. It employs separate buffers for sub-directories and data buckets, enabling different buffering polices that rely upon the page type (directory/data). At the moment, we utilize LRU as eviction policy in both buffers. The dirty evicted pages are not persisted immediately; they are accumulated into write buffers instead, enabling batch writes that accelerate performance.

We also examine a special case of H-Grid (Fig. 2), where all sub-directories are placed to the 3DXPoint storage, along with a number of selected data buckets. This approach can provide additional performance gain, since the sub-directories are referenced more frequently and their access pattern usually involves small-size I/O. The induced space overhead is not prohibitive since, as experimental results indicate, the size of directory pages in the physical storage is two orders of magnitude less than that of data buckets. The algorithms in the rest of the document were modified accordingly to comply with this special case.

In the sequel, we describe the Hybrid Bucket detection algorithm and the Search/Insert operations in H-Grid.

3.2 Hot Region Detection Algorithm

The role of the hybrid bucket detection algorithm is to reveal the most important, from performance viewpoint, sub-directories and data buckets. Only these will be migrated to the 3DXPoint storage. We use a temperature-based model to identify hot spatial regions that impose the highest I/O cost. These regions are represented by a number of sub-directories and data buckets. The weight of a sub-directory is highly correlated with the number of previous requests for it.

Equation 1 provides a metric for the weight W_i^σ of a certain sub-directory i.

$$W_i^\sigma = F_i^\sigma - \left(1 - \frac{t_i^\sigma}{T_i^\sigma}\right) \tag{1}$$

The first term expresses the frequency of accesses to the specific sub-directory, normalized into the range [0,1]. The second term refers to an aging policy, providing an advantage to sub-directories that were recently accessed. T_i^σ is the current timestamp, while t_i^σ is the timestamp of the previous access of sub-directory i. In other words, the second term reflects the changes occurring in the access patterns over time.

Regarding data buckets, we use a similar policy, as expressed by Eq. 2.

$$W_j^\beta = (W_i^\sigma + F_j^\sigma) - \left(1 - \frac{t_j^\beta}{T_j^\beta}\right) \tag{2}$$

Specifically, we utilize the number of read requests F_j for the bucket j and the weight W_i^σ of its parent sub-directory i to determine its eligibility. The aging factor is also applied to decay the weight of buckets that are rarely used. An additional condition for the data buckets is the presence of their parent sub-directory in the 3DXPoint storage as well.

The selection Algorithm (Alg. 1) uses the weight values to identify the hottest buckets. Only these are migrated to the 3DXPoint storage. The algorithm initially calculates the weight of a bucket (lines 1-3). In the sequel, it uses the cumulative moving average (CMA) of the weights (line 7) to determine the bucket's eligibility

```
Algorithm 1: HybridBucketDetect($B, S, WS$)
   Data: the bucket $B$, the parent sub-directory $S$, the weight of
         the parent sub-directory $WS$
   Result: Bucket is set hybrid or not
1  $F \leftarrow getBucketStats(B.id)$;
2  $D \leftarrow (T - B.t)/T$;
3  $W \leftarrow WS + F - D$;
4  $W_{SUM} \leftarrow W_{SUM} + W$;
5  $++n$;
6  if $S$ not in 3DXPoint then
7      return 0;
8  end
9  $CA \leftarrow W_{SUM}/n$;
10 if $W > s * CA$ then
11     $B.setHybrid \leftarrow 1$;
12     HBT[B.id]$\leftarrow W$;
13     set the 3DXPoint dirty flag of $B$;
14     return 1;
15 end
16 return 0;
```

```
Algorithm 2: Search($p, S, WS$)
   Data: the search point $p$, the parent Sub-directory $S$, the
         weight $WS$ of the parent sub-directory
   Result: the bucket $B$ wherein $p$ is located
1  search the scales to convert the coordinates of $p$ into interval
     indexes;
2  use interval indexes to locate bucket $B$ in the sub-directory;
3  FetchBucket($B$);
4  if $HBT[B.id]$ is not $NULL$ then
5      update $HBT[B.id]$ with new weight value;
6  else
7      HybridBucketDetect($B$,$WS$);
8  end
9  update $B$ timestamp;
10 return $B$;
```

```
Algorithm 3: FetchBucket($B, HBT, MB$)
   Data: the id of bucket $B$ to be read, the Hybrid Bucket Table
         $HBT$, in-memory buffer MB
   Result: the bucket $B$
1  if $B$ is in main memory buffer $MB$ then
2      move $B$ to the MRU position of main buffer;
3  else if $B$ is in flash SSD write buffer then
4      move $B$ to the MRU position of main buffer;
5  else if $B$ is in 3DXPOINT SSD write buffer then
6      move $B$ to the MRU position of main buffer;
7  else if $HBT[B.id]$ is not $NULL$ then
8      read $B$ from 3DXPOINT SSD;
9      move $B$ to the MRU position of main buffer;
10 else
11     read $B$ from flash SSD;
12     move $B$ to the MRU position of main buffer;
13 end
14 return $B$;
```

for the 3DXPoint storage. The simple moving average (SMA) in sequential time windows can be used alternatively. Upon a hot bucket is detected, a dirty flag is set, forcing the bucket to be written on the 3DXPoint SSD during the next write-buffer flush (line 13). The parameter s is a tunable constant which controls the selectivity of the algorithm. *HBT* (Hybrid Bucket Table) is a hash table that maps all buckets in the 3DXPoint storage to their respective weights. The weight value of a bucket in the *HBT* is updated every time the bucket is retrieved. This algorithm is adapted for the sub-directories as well.

3.3 Single Point Search

In the two-level Grid-File, the search operation starts from the in-memory root directory by locating the sub-directory which contains a particular point. When the sub-directory is retrieved, the procedure continues at the sub-directory level, looking for the appropriate bucket. In this way, the Grid-File guaranties that a single point is reached in two disk accesses.

In H-Grid the search operation is adjusted to the hybrid storage configuration. Algorithm 2 describes the operation for a given point p at sub-directory level, while it adapts similarly at the root level. Initially, the linear scales and the grid are used to find out the address of bucket B that contains p. A fetch operation for B is issued either to flash or to the 3DXPoint storage (line 3). If B already resides in the 3DXPoint, its weight value W_B^β is updated. Otherwise, Algorithm 1 is employed to decide if B is eligible for migration (lines 4-8). Finally, the last access timestamp of B is updated and B is returned.

Algorithm 3 details the bucket fetching operation in H-Grid. If the requested bucket B is already in the in-memory buffer (*MB*) or into one of the two write buffers (flash or 3DXPoint), then B is moved to the most recently used (MRU) position of *MB*. Otherwise, a fetch operation from the secondary storage is initiated. The *HBT* table is examined and a bucket read request is issued to the appropriate storage device. By the end of the operation, B is placed to the *MRU* position of the main buffer and a reference to it is returned.

From the above, it is obvious that the two disk access principle of Grid File is also preserved in H-Grid. The cost of searching a single point in H-Grid is determined by the cost of retrieving the directory and bucket pages from the physical storage.

Thus, for a given search query Q, let $x_s \in \{0, 1\}$ represent whether sub-directory s is stored in the 3DXPoint storage or not, and $x_b \in \{0, 1\}$ denote whether the bucket b is in 3DXPoint or not as well. The cost C_Q of Q is

$$\begin{aligned} C_Q &= x_s * R_x + R_f * (1 - x_s) + x_b * R_x + R_f * (1 - x_b) \\ &= 2 * R_f - (R_f - R_x) * (x_s + x_b) \end{aligned}$$

where R_f and R_x denote the cost of reading a page from the flash and 3DXPoint, respectively. The wider the difference in page read time between flash and 3DXPoint gets, the higher the performance gain of H-Grid becomes.

3.4 Insert Point

Algorithm 4 describes the insertion of a new point to the H-Grid. It receives as input a point p and exploits the *Search* operation to acquire the bucket B wherein p has to be inserted. If B is not full, a proper record is composed and is added to it (lines 2-14). In case B resides in the flash SSD, the hybrid bucket detect operation is invoked, testing its eligibility for migration to the 3DXPoint storage. A proper dirty flag is set denoting bucket's storage medium. This flag is exploited by the write operation. Each bucket B accommodates a certain number of records. In case B is full, a split operation of B is initiated, resulting in the introduction of a new bucket. Successive insertions of new records may cause a sub-directory split as well.

Algorithm 4: Insert(p, S, WS)

```
Data: the new entry $p$ to be inserted, the parent sub-directory
      $S$, the weight of the parent sub-directory $WS$
1  $B \leftarrow Search(p, S, WS)$
2  if $B$ is not full then
3      insert record ($p$) to $B$;
4      if $HBT[B.id]$ is not $NULL$ then
5          set the 3DXPoint dirty flag of $B$;
6      else
7          if not $HybridBucketDetect(B,S,WS)$ then
8              set the flash dirty flag of $B$;
9          end
10     end
11     update $B$ timestamp;
12     return 1;
13 else
14     split bucket $B$;
15     Insert($p, S, WS$)
16 end
```

4 PERFORMANCE EVALUATION

4.1 Methodology and setup

In this section we present the evaluation of H-Grid using both flash and 3DXPoint storage devices. We present the performance benefits of H-Grid against flash efficient (GFFM [5]) and traditional (R*-Tree [7]) spatial indexes that are unable to exploit diverse storages. We also test the special case of H-Grid, presented in Section 3.1, that persists all its sub-directories to the 3DXPoint storage.

All the experiments were performed on a workstation running CentOS Linux 7 (Kernel 4.14.12). The workstation is equipped with a quad-core Intel Xeon CPU E3-1245 v6 3.70GHz CPU, 16GB of RAM, and a SATA SSD for hosting the operating system. The experiments were conducted on an INTEL DC P3700 480GB PCI-e 3.0 SSD (FLASH) and an Intel Optane 32GB Memory Series device (3DXPoint). The latter belongs to the first generation of devices utilizing 3DXPoint memory. Table 1 summarizes the performance characteristics of the two devices as provided by manufacturers' data sheets.

Table 1: SSD Characteristics

	Intel DC P3700	Optane Memory series
	(Flash)	(3DXPoint)
Seq. Read	up to 2700MB/s	up to 1350MB/s
Seq. Write	up to 1100MB/s	up to 290MB/s
Random Read	450K IOPS	240K IOPS
Random Write	75K IOPS	65K IOPS
Latency Read	120μs	7μs
Latency Write	30μs	18μs

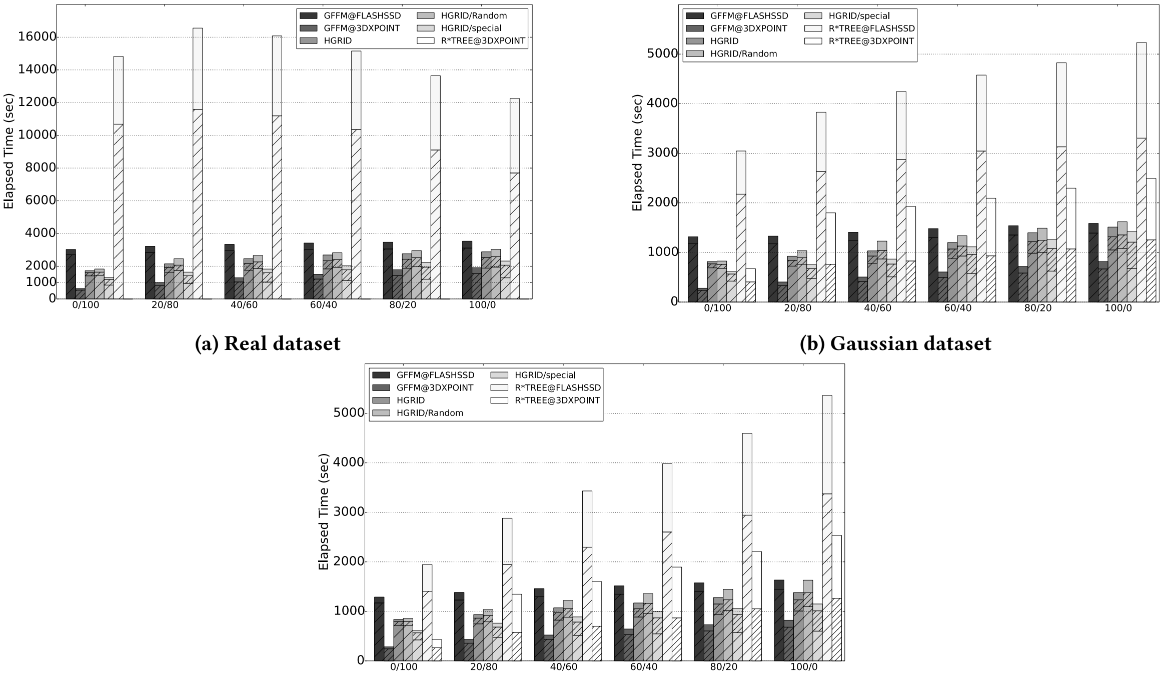
We use two synthetic and one real dataset for the experiments. The synthetic datasets follow Gaussian and Uniform distributions, respectively, while the real one contains geographical points extracted from Openstreetmap[1]. All experiments were executed using the Direct I/O (O_DIRECT) option to bypass the Linux OS caching system. We varied the selectivity parameter s (Alg. 1) in the range 1.0 to 2.5 in the various workloads. In this way, a number of up to 40% of the sub-directories and up to 20% of data buckets migrated to 3DXPoint. We set the total size of the in-memory buffers for every examined index to 8MB. We did not manage to run R*-tree on the 3DXPoint using the real dataset due to lack of space.

4.2 Insert/Search Queries

We evaluated the performance of H-Grid using six different workloads for each dataset. Regarding the real dataset, the indexes were initialized with 500M points. Figure 3a presents the elapsed time for 10M operations with the specified search and insert ratios. Specifically, H-Grid achieves a speedup which ranges from 18.4% to 43% in comparison to the execution of GFFM on the flash SSD (baseline). H-Grid does not provide adequate results when the buckets for the 3DXPoint are randomly selected. This fact reveals the efficiency of the proposed hot region detection algorithm. The special case of H-Grid, which considers placing all sub-directories in the 3DXPoint storage, achieves a significant performance gain that ranges from 34.4% to 56.6%. The acquired results are even better when GFFM exclusively utilizes the 3DXPoint SSD as persistent storage (best case), providing a speedup reaching 78.9% in comparison to the execution on the flash SSD.

For the synthetic dataset runs, we used 50M points for initialization and 5M I/S operations for testing. As depicted in Figures 3b and 3c, there is remarkable improvement in all experiments involving read sensitive workloads. Figure 3b presents the results for the Gaussian dataset. Specifically, using the 3DXPoint SSD as sole storage medium for GFFM, we achieve an improvement ranging from 49.7% to 77.9% comparing with its execution on the flash one. H-Grid achieves a performance gain up to 35% in comparison to the GFFM run on the flash SSD. The special case of H-Grid exhibits even better performance (29.6%-52.7%) as expected. The results are similar in the test cases that utilize the uniformly distributed dataset (Fig. 3c). The H-Grid is up to 35% faster than the GFFM execution on the flash SSD, while the special case improves further

[1] http://spatialhadoop.cs.umn.edu/datasets.html

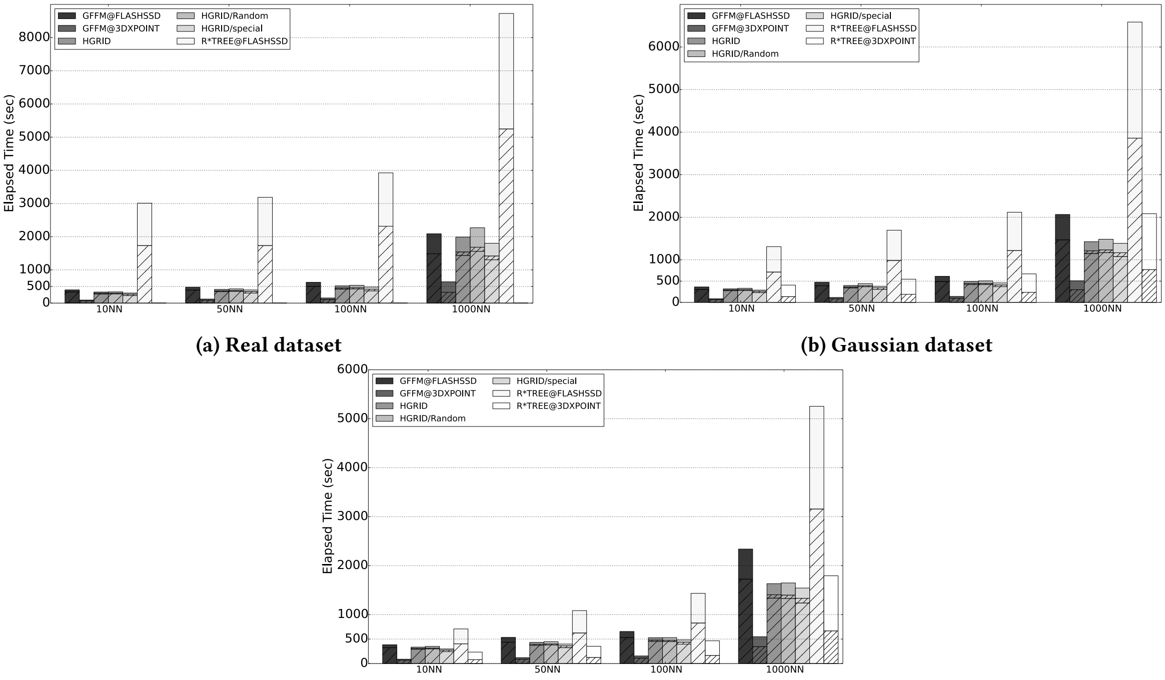


(a) Real dataset

(b) Gaussian dataset

(c) Uniform dataset

Figure 3: Execution times of I/S queries for different workloads. H-Grid provides better results when searches are the majority.



(a) Real dataset

(b) Gaussian dataset

(c) Uniform dataset

Figure 4: Execution times of kNN queries for different workloads.

the result. The low-latency of 3DXPoint SSD also contributes significant performance gains for R*-Tree comparing with its flash-based execution.

4.3 kNN Queries

In this section we analyze the performance of kNN queries. We previously initialized the indexes, using the same datasets as in I/S queries (500M points for the real dataset, 50M points for each synthetic). Figure 4a depicts the results of the real dataset, while Figures 4b and 4c correspond to the Gaussian and Uniform test cases. Regarding the real dataset, H-Grid provides a gain up to 17% in the 100NN case, while in the 1000NN case the gain is only 4.7%. This is due to the large number of bucket reads that imposes. The results are better in the smaller synthetic workloads. Particularly, for the Gaussian dataset, the improvement ranges from 12.3% for the 10NN query, and up to 31% for the 1000NN one. Similarly, in the experiments with the Uniform dataset, a speedup ranging from 12.4% up to 30% is achieved. H-Grid achieves better results when all the sub-directories reside in the 3DXPoint (special case). Specifically, for the real dataset, it improves its execution time starting in a range from 13.6% up to 23.5%.

4.4 Range Queries

We discuss the performance of range queries next. Specifically, we present the elapsed times of 5K requests issued to each one of the examined indexes. We posed 1M queries to the previously initialized indexes. Figure 5 summarizes the results for all test cases. H-Grid improves GFFM on flash SSD (baseline case) up to 28% in the real dataset run, while the gain for H-Grid is smaller in the runs that use the synthetic data. The efficiency of the proposed hot region detection algorithm is proven to be true once again, since the random selection of buckets for migration leads to worse results. The sole execution of GFFM in the 3DXPoint SSD provides significant performance improvements which range from 74.5% to 78%. Remarkable is also the speedup for the R*-tree (75%), when it utilizes the 3DXPoint SSD, for the Gaussian and Uniform workloads.

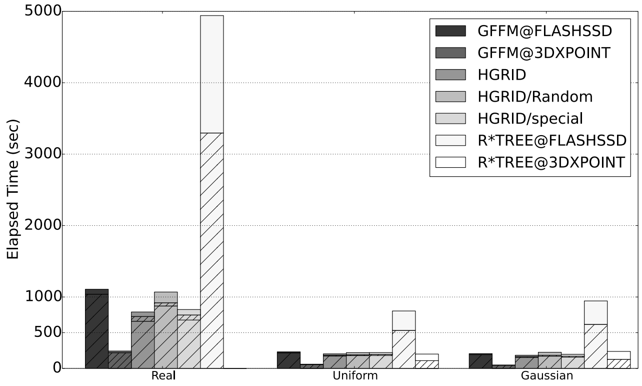


Figure 5: Execution times of range queries for three different datasets.

5 CONCLUSIONS

In this paper we put effort to highlight the opportunities that new or upcoming non-volatile memory technologies create for data indexing. We studied the performance of spatial indexes exploiting 3DXPoint NVM as secondary storage and we introduced H-Grid, a spatial index for hybrid storage. H-Grid detects hot regions and persists them in 3DXPoint storage.

The experimental results show significant performance improvement for H-Grid in comparison to GFFM, a flash-based Grid File variant. Specifically, the gain ranges from 35% up to 43% in the single point retrieval, while the achieved speedup for range and kNN queries is up to 28% and 32%, respectively. Examining the attained results from H-Grid, we can infer that tree indexes, like B-trees and R-trees can also benefit significantly by storing hot nodes to the low-latency 3DXPoint. This gain can be higher in workloads that impose small random I/O.

So, we demonstrated that even small amounts of 3DXPoint in the secondary storage layer can accelerate spatial queries performance at affordable cost (e.g. a 32GB Optane module costs under 100 USD). Our plans for future work in H-Grid include a method for tuning the selectivity parameter *s* based on workload's characteristics and a cooling process for buckets that stay long time in the 3DXPoint storage without being accessed. We also intend to study the performance characteristics of tree indexes, like B-trees and R-trees, in non-volatile storage.

REFERENCES

[1] D. Agrawal, D. Ganesan, R. Sitaraman, Y. Diao, and S. Singh. Lazy-adaptive tree: An optimized index structure for flash devices. *Proceedings of the VLDB Endowment*, 2(1):361–372, 2009.

[2] M. Canim, G. A. Mihaila, B. Bhattacharjee, K. A. Ross, and C. A. Lang. Ssd bufferpool extensions for database systems. *Proceedings of the VLDB Endowment*, 3(1-2):1435–1446, 2010.

[3] A. C. Carniel, R. R. Ciferri, and C. D. de Aguiar Ciferri. A generic and efficient framework for spatial indexing on flash-based solid state drives. In *Advances in Databases and Information Systems*, pages 229–243. Springer, 2017.

[4] F. Chen, D. A. Koufaty, and X. Zhang. Hystor: making the best use of solid state drives in high performance storage systems. In *Proceedings of the international conference on Supercomputing*, pages 22–32. ACM, 2011.

[5] A. Fevgas and P. Bozanis. Grid-file: Towards to a flash efficient multi-dimensional index. In *International Conference on Database and Expert Systems Applications*, pages 285–294. Springer, 2015.

[6] A. Fevgas and P. Bozanis. Lb-grid: An ssd efficient grid file. *Data Knowledge Engineering*, 2019 in press.

[7] M. Hadjieleftheriou. libspatialindex 1.8.5, 2019. [Online; accessed 20-Feb-2019].

[8] F. T. Hady, A. Foong, B. Veal, and D. Williams. Platform storage performance with 3d xpoint technology. *Proceedings of the IEEE*, 105(9):1822–1833, 2017.

[9] K. Hinrichs. Implementation of the grid file: Design concepts and experience. *BIT Numerical Mathematics*, 25(4):569–592, 1985.

[10] P. Jin, X. Xie, N. Wang, and L. Yue. Optimizing r-tree for flash memory. *Expert Systems with Applications*, 2015.

[11] P. Jin, C. Yang, C. S. Jensen, P. Yang, and L. Yue. Read/write-optimized tree indexing for solid-state drives. *The VLDB Journal*, 25(5):695–717, 2016.

[12] P. Jin, P. Yang, and L. Yue. Optimizing b+-tree for hybrid storage systems. *Distributed and parallel Databases*, 33(3):449–475, 2015.

[13] I. Koltsidas and V. Hsu. IBM Storage and NVM express Revolution. Technical report, IBM, 2017.

[14] G. Li, P. Zhao, L. Yuan, and S. Gao. Efficient implementation of a multi-dimensional index structure over flash memory storage systems. *The Journal of Supercomputing*, 64(3):1055–1074, 2013.

[15] L. Li, P. Jin, C. Yang, S. Wan, and L. Yue. Xb+-tree: A novel index for pcm/dram-based hybrid memory. In *Australasian Database Conference*, pages 357–368. Springer, 2016.

[16] Y. Li, B. He, Q. Luo, and K. Yi. Tree indexing on flash disks. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 1303–1306. IEEE, 2009.

[17] S. Lin, D. Zeinalipour-Yazti, V. Kalogeraki, D. Gunopulos, and W. A. Najjar. Efficient indexing data structures for flash-based sensor devices. *ACM Transactions*

on Storage (TOS), 2(4):468–503, 2006.

[18] J. D. Little and S. C. Graves. Little's law. In *Building intuition*, pages 81–100. Springer, 2008.

[19] X. Liu and K. Salem. Hybrid storage management for database systems. *Proceedings of the VLDB Endowment*, 6(8):541–552, 2013.

[20] Y. Liu, X. Ge, X. Huang, and D. H. Du. Molar: A cost-efficient, high-performance hybrid storage cache. In *Cluster Computing (CLUSTER), 2013 IEEE International Conference on*, pages 1–5. IEEE, 2013.

[21] Y. Lv, X. Chen, G. Sun, and B. Cui. A probabilistic data replacement strategy for flash-based hybrid storage system. In *Asia-Pacific Web Conference*, pages 360–371. Springer, 2013.

[22] R. Micheloni. *3D Flash memories*. Springer, 2016.

[23] J. Nievergelt, H. Hinterberger, and K. C. Sevcik. The Grid file: Aan adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems*, 9(1):38–71, 1984.

[24] J. Niu, J. Xu, and L. Xie. Hybrid storage systems: A survey of architectures and algorithms. *IEEE ACCESS*, 6:13385–13406, 2018.

[25] H. Roh, S. Kim, D. Lee, and S. Park. As b-tree: A study of an efficient b+-tree for ssds. *Journal of Information Science and Engineering*, 30(1):85–106, 2014.

[26] H. Roh, S. Park, S. Kim, M. Shin, and S.-W. Lee. B+-tree index optimization by exploiting internal parallelism of flash-based solid state drives. *Proceedings of the VLDB Endowment*, 5(4):286–297, 2011.

[27] H. Roh, S. Park, M. Shin, and S.-W. Lee. Mpsearch: Multi-path search for tree-based indexes to exploit internal parallelism of flash ssds. *IEEE Data Eng. Bull.*, 37(2):3–11, 2014.

[28] G. Roumelis, A. Fevgas, M. Vassilakopoulos, A. Corral, P. Bozanis, and Y. Manolopoulos. Bulk-loading and bulk-insertion algorithms for xbr-trees in solid state drives. *Computing*, 2019. available online.

[29] H. Samet. Applications of spatial data structures. Addison-Wesley, 1990.

[30] M. Sarwat, M. F. Mokbel, X. Zhou, and S. Nath. Fast: a generic framework for flash-aware spatial trees. In *Advances in Spatial and Temporal Databases*, pages 149–167. Springer, 2011.

[31] J. Yang and D. J. Lilja. Reducing relational database performance bottlenecks using 3d xpoint storage technology. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1804–1808. IEEE, 2018.

[32] J. Zhang, M. Kwon, D. Gouk, S. Koh, C. Lee, M. Alian, M. Chun, K. N. Kandemir MTaylan, J. Kim, and M. Jung. Flashshare: Punching through server storage stack from kernel to firmware for ultra-low latency ssds. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 477–492, 2018.

Mining Complex Temporal Dependencies from Heterogeneous Sensor Data Streams

Amine El Ouassouli
Univ Lyon, INSA Lyon, Foxstream
LIRIS (UMR 5205 CNRS)
Villeurbanne, France
amine.el-ouassouli@insa-lyon.fr

Lionel Robinault
Foxstream
LIRIS (UMR 5205 CNRS)
Vaulx-En-Velin, France
l.robinault@foxstream.fr

Vasile-Marian Scuturici
Univ Lyon, INSA Lyon
LIRIS (UMR 5205 CNRS)
Villeurbanne, France
marian.scuturici@insa-lyon.fr

ABSTRACT

In addition to sensor heterogeneity, monitoring applications must handle different temporal data models (e.g time series, event sequences). In this paper, we address the problem of discovering directly actionable high level knowledge from such data. We model temporal information through interval-based streams describing environment states. We propose an approach to discover efficiently Complex Temporal Dependencies (CTD) between state streams, called CTD-Miner. A CTD is modeled similarly to a conjunctive normal form and describes temporal relations (time delays) between states. CTD-Miner is robust to temporal variability of data and uses a statistical independence test to determine the most appropriate time lags between states. This test is also used to perform pruning on sub-dependencies checking. Finally, we validate our approach via synthetic data and a case study in a real-world smart environment using outdoor cameras and real-time video processing.

CCS CONCEPTS

• **Information systems** → **Data streams**; **Temporal data**; **Sensor networks**; **Data analytics**; **Data stream mining**.

KEYWORDS

Data streams, temporal data, mining complex events

ACM Reference Format:
Amine El Ouassouli, Lionel Robinault, and Vasile-Marian Scuturici. 2019. Mining Complex Temporal Dependencies from Heterogeneous Sensor Data Streams. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3331076.3331112

1 INTRODUCTION

Temporal knowledge discovery is an important task for a growing number of application domains where large volumes of timestamped data can be generated. Smart environments are a typical example of such contexts. They refer to places or objects equipped with a sensor system monitoring one or more physical measures through data streams. These make it possible to obtain temporal description of the environment's characteristics evolution that is induced by temporal phenomena occurring within it. In this context, a knowledge discovery task consists in extracting non-trivial, expressive and concise patterns describing typical hidden temporal phenomena.

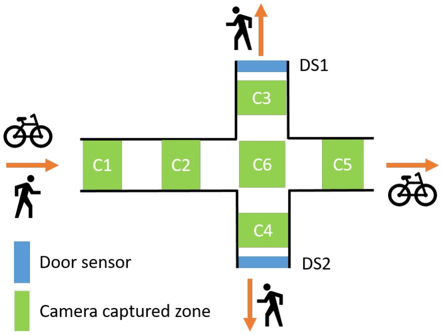


Figure 1: A smart environment

We describe in Fig. 1 several one-way corridors used by actors of two types: pedestrians and cyclists. This environment is equipped with a sensor system composed of door sensors and video cameras monitoring parts of the corridor. For data produced by video cameras, advances in image and video processing make it possible to obtain useful insights: motion detection, counts, object recognition. In this example, C1, C3, C4 and C5 provides motion detection, C2 provides object recognition and C6 counts moving objects. Each of these sensors provides a data streams to a monitoring application. Examples are shown in Fig.2. Our objective is to extracts temporal knowledge from such configurations.

Temporal descriptions of sensed environment are richer if different types of features are used. In the example described higher,

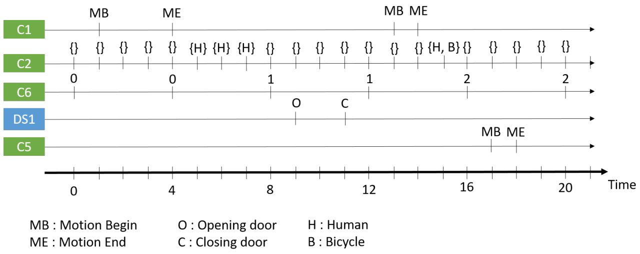


Figure 2: Some raw data streams gathered from the sensor system depicted in Fig.1

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331112

the analysis of motion information would permit to obtain 3 main trajectories *"C1 then C3"*, *"C1 then C4"* and *"C1 then C5"*. In this example, C3 and C4 can only be reached by pedestrians and C5 by cyclists. A more valuable insight is given by the heterogeneous relation *"C1 then <Bicycle> then C5"*. The usage of heterogeneous descriptions for sensed environments offers the opportunity to discover temporal knowledge reporting on complex relations between different features of the environment. This poses a challenging problem to temporal pattern mining approaches:

How to obtain complex, non-trivial and directly actionable temporal knowledge starting from heterogeneous sensor raw data?

Each sensor may use the most appropriate time model (time points or intervals) and data model (e.g numeric time series, symbolic events or item sets) w.r.t its physical measure. A simple way to address heterogeneity is to transform raw sensor data sequences to a unified general model using a Temporal Abstraction (TA) operation. In [9] authors defined TA as "*the segmentation and/or aggregation of series of raw (...) data into a symbolic time-interval series representation, often at a higher level of abstraction (...), suitable for human inspection or for data mining*". In addition to solving heterogeneity problems, TA permits to build high level *pattern vocabulary* that it more suitable for human perception and interpretation [5]. In this work, we use an interval-based representation built on states referring to data configurations of interest for the application domain. States are defined via predicates on one or more sensors producing data. A state stream contains parts of time (intervals) where the state's predicate is valid. This way, data provided by a sensor system is transformed to a set of unified interval-based state streams.

Time intervals allow to take into account straightforwardly duration in a discrete time contrary to point based events. Therefore, using a point-based approach to process interval data induces a loss of information and expressiveness. As discussed in [1], 13 relations can exist between two intervals. Existing interval-based qualitative pattern models, based on all or a subdivision of Allen' logic, may suffer from various expressiveness issues as ambiguity (a pattern may lead to different temporal relations) or completeness (capability of expressing all possible relations) [5]. In some extent, quantitative patterns maintaining temporal information permit to solve these problems as temporal relations are explicitly characterized permitting to infer Allen' logic. Moreover, time lags can also be a discriminant factor. In the example described higher, the trajectory *"C1 then C2"* can be performed by pedestrians and cyclists. These two types of actors performs the same qualitative trajectory but with different temporal information: cyclists *"C1 then C2 after d units of time"* are faster than pedestrians *"C1 then C2 after d' units of time"*, where $d < d'$. This can be often useful, for example, to perform forecasting: *"if C1 then C2 after d"*, one can predict that the following trajectory step is C5 (rule corresponding to cyclists).

Very few existing approaches tackled directly quantitative interval patterns in streams or can easily be adapted to this task ([11], [4]). In [11], authors introduced a novel form of temporal relations based on the assessment of intervals intersection. The intersection of a pair of state streams contains time portions where both states are active. The length of this intersection is used to assess statistically the significance of the temporal correlation. In our work, this model is extended to handle more complex relations involving multiple states. We want to express relations including disjunctions and conjunctions involving multiple data streams, like *"IF A then (B or C) after a duration d"*.

In this work, we firstly introduce the Complex Temporal Dependency model (CTD), a quantitative pattern model as an extension of pairwise dependencies proposed in [11]. It is automatically assessed using a χ^2 test of independence on interval intersection length. Next, we propose CTD-Miner, an algorithm devised to discover efficiently CTDs. It includes a novel time lag discovery method, sub-dependencies pruning techniques and dependencies merging. Finally, we conducted experiments using both simulated and real life motion sensor data to validate our approach.

2 RELATED WORK

Sequential and temporal pattern mining is a well studied research area designed to discover regularities among temporally ordered data. Contributions in this field can be categorized following three main criteria: data format (transaction databases or streams/single sequences), time models (time points or intervals) and temporal description (qualitative or quantitative).

Most of existing approaches consider input data to be in a transactional format: each transaction is a temporally ordered sequence associated with an ID. A typical example would be medical data: a sequence describes medical history of a particular patient identified by an ID. Transactional format makes clear separations between activities (e.g between patients medical records) and can be called "subject-centered". With this data format, temporal pattern mining task consists on finding temporal regularities between transactions. On the other hand, sensor streams (or single sequences) are a continuous flow of data where no boundary exists between activities and describe the evolution of physical measures. For example, a motion sensor reports on motion rather than describing motion of objects separately: a same piece of data can be generated by one or various actors activity. Hence, sensor data can be called "measure-centered". In both cases, existing contributions use a user-given minimum support referring to the part of data where a relation stands. The support assessment can be based on number of transaction for transactions, time windows [7] or number of items [13].

Most of the contributions dealing with streaming data or single sequences focus on time points with qualitative patterns reporting on *before/after/co-occurring* relations [7]. Quantitative patterns do not permit a time delay discrimination which is rather useful for many application domains. While some contributions integrate temporal constraints (time windows [7], gap constraints [2]), several approaches tackled quantifying time delays ([6] [13] [14]). Another category of contributions deals with data represented as intervals. As discussed in [1] this time model permits to express more complex relations (13 relation types in Allen' logic). This form of patterns may suffer from expressiveness limitations as ambiguity (a pattern may lead to different relations) or completeness (not allowing to express all possible relations). We refer to [5] for a detailed analysis of expressiveness. Besides, qualitative patterns are sensitive to temporal variability as slight variation in intervals endpoints may lead to different qualitative relations.

Quantitative patterns permit to deal with these problems as they include temporal information permitting to infer Allen' logic.

Among the existing contributions [10] [3] [12] only two recent approaches are designed or can be easily adapted to interval streams [4] [11]. In [4], the authors propose PIVOTMiner that uses a geometric approach consisting on the projection of each interval $[t_{begin}, t_{end})$ into a bi-dimensional plane ($begin, end$) where the temporal relation between two intervals is considered as a vector. This allows to perform a DBSCAN clustering after an origin transformation stage: to mine relations of type $A \longrightarrow B$, all vectors with A as source and B as target are moved such as all sources coincides with the space origin. Cluster centroides provides the time lag information. While designed for sequence databases, this approach can be easily adapted to data streams as it is not endpoint sensitive (as in [12], [3]). Finally, in [11] the authors proposed a novel form of temporal relations made possible by the usage of intervals. It is based on the intersection of sets of validity intervals corresponding to predicates describing the environment states. The authors propose an algorithm, TEDDY, devised to discover dependencies of type *"A ⟶ B after* (α, β) *units of time"* assessed via a confidence measure.

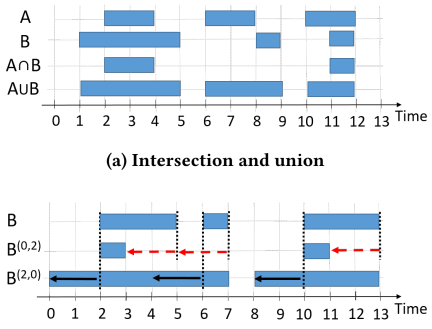
3 STATE STREAMS

3.1 From raw sensor data to state streams

We define a data stream D as a sequence of time stamped data produced by a source $d \in \Lambda$, with Λ the set of data sources composing a sensor system. Formally, a data sequence S_d produced by d is defined with $S_d = \{\langle (t, v)\rangle \mid t \in \mathcal{T}, v \in \mathcal{V}_d\}$. $\mathcal{T}$ is an infinite set of discrete time stamps and t can be either a time point or an interval $t = [t_b, t_b + 1)$. $\mathcal{V}$ is the set of possible values given by d. We assume that a data source cannot produce more than one value at a time. In the example depicted in Fig 1, the set of data sources is $\Lambda = \{C1, C2, ...DS1, DS2\}$. Possible values for the data source $C1$ is a set of event labels $\mathcal{V}_{C1} = \{MB, ME\}$, for $C2$ is all subsets of possible objects $\mathcal{V}_{C2} = \{O_i \mid O_i \subseteq \{H, B\}\}$, and for $C6$ is the set of positive integers $\mathcal{V}_{C6} = \mathbb{N}$. Examples of state streams are provided in Fig 2. Notice that data sources can be heterogeneous at different levels: sample rates, data models (e.g time series, labeled events, item sets), time models (time points or intervals). Our objective is to process data streams given by Λ in order to obtain high level temporal knowledge.

Our approach to solve problems introduced by heterogeneity is to use Temporal Abstraction (TA) devised to transform raw heterogeneous data into a unified interval based model using high level abstraction providing a "pattern vocabulary" for temporal knowledge. This technique was often applied for medical records (e.g [9]) or with sensor time series (e.g [8]). More precisely, the main idea is to use expert knowledge or automatic techniques (e.g time series discretization) to define a batch of high level states that can be seen as "environment features" of interest. A state is a particular environment configuration referring to an non-trivial situation. Formally speaking, a state a is defined as a predicate: $a : \mathcal{T}, \Lambda \longrightarrow \mathbb{B}$. The result of a is a boolean value stating that the state is whether active or inactive at $[t, t+1) \in \mathcal{T}$ given a set of data sequences in Λ. Some examples of states:

- **motionInC1**$(t, C1)$ ::= $last(t, C1).v = MB$
- **increasingOccupency**$(t, C6)$ ::= $last(t, C6).v - last(t - 1, C6).v > 0$
- **congestion**$(t, C1, C6)$ ::= $(last(t, C1).v = MB) \wedge (last(t, C6).v - last(t-1, C6).v = 0)$



(a) Intersection and union

(b) (α, β)**-transformation**

Figure 3: Examples of intersection and union and (α, β)**-transformation**

Predicates can be simple conditions on a single raw data sequence values as *motionInC1* that reports on simple motion activity (the last event produced by $C1$ at the timestamp t is MB=Motion Begin). States can also report on data trends as *increasingOccupency*, or can also integrate data from various sensors as for *congestion.*

A state stream is a temporally ordered sequence containing all time intervals, called active intervals, where its corresponding state predicate is verified. A state stream A corresponding to state a is formally defined as $A = (a(t, \lambda), \langle [t_{b_i}, t_{e_i})\rangle)$ such that $\forall t_{b_i}, t_{e_i} \in \mathcal{T}, t_{b_i} < t_{e_i} < t_{b_{i+1}}$ and $\forall t \in [t_{b_i}, t_{e_i}) \mid a(t, \lambda) = True$ with $\lambda \subseteq \Lambda$. The size of a state stream A noted $\#A$ corresponds to the number of its intervals. The length of a state stream is the sum of its active intervals duration:

$$len(A) = \sum_{[t_b, t_e) \in A} (t_e - t_b)$$

For example, the state stream corresponding to $motionInC1$ from Fig. 2 is $motionInC1 =< [1, 4), [13, 14) >$, with $size(motionInC1) = 2$ and $len(motionInC1) = 4$.

3.2 Operations on state streams

We define several operations on state streams: intersection, union and temporal transformation.

The intersection of two state streams A and B, noted $A \cap B$ is a state stream containing intervals where both A and B are active (Fig.3). Formally, $A \cap B = (a \wedge b, \langle [t_{b_i}, t_{e_i})\rangle$ such that $\forall t \in [t_{b_i}, t_{e_i}), \exists [t_{b_j}, t_{e_j}) \in A, [t_{b_k}, t_{e_k}) \in B$ such that $t \in [t_{b_j}, t_{e_j})$ and $t \in [t_{b_k}, t_{e_k})$. This operation is computed in $\Theta(Max(\#A, \#B))$.

The union of two state streams A and B, noted $A \cup B$, produces a new state stream containing the intervals where A or B are active (Fig. 3). Formally, $A \cup B = (a \wedge b, \langle [t_{b_i}, t_{e_i})\rangle$ such that $\forall t \in [t_{b_i}, t_{e_i}), \exists [t_{b_j}, t_{e_j}) \in A, [t_{b_k}, t_{e_k}) \in B$ such that $t \in [t_{b_j}, t_{e_j})$ or $t \in [t_{b_k}, t_{e_k})$. Similarly to intersection, the union of two state streams has a time complexity of $\Theta(Max(\#A, \#B))$.

A state stream B can also be temporally shifted via an (α, β)-transformation. This operation results on a state stream $B^{(\alpha, \beta)} = \langle [t_{b_i} - \alpha, t_{e_i} - \beta) \mid [t_{b_i}, t_{e_i}) \in B\rangle$. Hereafter, α is called expansion and

β reduction. We describe in Fig. 3 two examples: a $(0, 2)$-reduction $B^{(0,2)}$ and a $(2, 0)$-expansion $B^{(2,0)}$. This temporal transformation is done in $\Theta(\#B)$.

4 COMPLEX TEMPORAL DEPENDENCIES

This section introduces the Complex Temporal Dependencies (CTD) model that aims to describe temporal correlations between multiple state streams on the basis of their intersection length.

4.1 Background

A pairwise temporal dependency between state streams A and B, noted $A \longrightarrow B$, notifies that A *occur simultaneously with* B. We call hereafter A the premise and B the conclusion. $A \longrightarrow B$ is assessed with the intersection length of A and B via the following confidence measure:

$$conf(A \longrightarrow B) = \frac{len(A \cap B)}{len(A)}$$

Notice that confidence is maximal (=1) if all active intervals of A are included in active intervals of B: A occurs *always* with B. For example, in Fig. 3 $conf(A \longrightarrow B) = \frac{3}{6}$.

In order to find relations of interest, and to avoid the utilization of an user-given threshold, this confidence measure is statistically assessed via a Pearson χ^2 test of independence. The independence hypothesis states that A and B are statistically independent within a duration $\mathcal{T}$. It relies on the following assumption: if active length of B is uniformly distributed in $\mathcal{T}$, there is no significant correlation between A and B. The given validity threshold on confidence is noted $th(len(A), len(B))$[1] and obtained for a significance level of 0.05 and 1 degree of freedom [11].

Simultaneity do not permit to express dependencies when it happens that B is time-delayed regarding A. The conclusion stream can be temporally shifted with an (α, β)-transformation to obtain a relation of type $A \longrightarrow B^{(\alpha,\beta)}$. Therefore, the associated confidence value is given by:

$$conf(A \longrightarrow B^{(\alpha,\beta)}) = \frac{len(A \cap B^{(\alpha,\beta)})}{len(A)}$$

A dependency $A \longrightarrow B^{(\alpha,\beta)}$ means that: B starts at most α time units after A and B finishes at least β time units after A. For simplicity purposes, we refer hereafter to this relation with: if A is active then $B^{(\alpha,\beta)}$ is active.

4.2 Conjunctive and disjunctive relations

As previously described, pairwise state dependencies are composed of a premise, a conclusion and an inference operator "⟶" that is associated with a confidence measure and describes simultaneity and succession relations. Our goal is to express dependencies with multiple states.

In Fig.4 pairwise dependencies are $A \longrightarrow B^{(2,3)}, conf = \frac{3}{5}$ and $B \longrightarrow C^{(3,2)}, conf = \frac{6}{9}$. In this example, C always follows a succession "A then B": a dependency *A followed by B then C* should have a confidence of 1. This example permits to emphasize that dependencies between multiple states must be able to express conditional relations like: *state C is correlated with state B* ***if*** *B is preceded by state A after a duration d.* This is made possible by the introduction of the conjunctive operator $\wedge$ that expresses simultaneity or succession, but contrary to the inference it is not associated with a confidence measure. A conjunction of two state streams A and B, noted $A \wedge B$, corresponds to the intersection $A \cap B$. This conjunctive stream can be seen as the stream representation of $A \longrightarrow B$ and be used to extend this dependency. In the higher example, we can extract the following conjunctive dependencies:

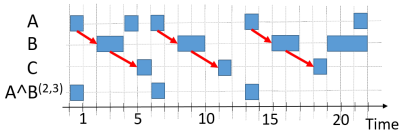


Figure 4: Example of 3 correlated streams

$$A \wedge B^{(2,3)} \longrightarrow C^{(5,5)}, conf = 1 \quad (1)$$

$$A \longrightarrow B^{(2,3)} \wedge C^{(5,5)}, conf = \frac{3}{6} \quad (2)$$

These two dependencies are to be interpreted differently: (1) If state A and $B^{(2,3)}$ are active **then** $C^{(5,5)}$ is active with a confidence of 1; (2) If A is active **then** $B^{(2,3)}$ and $C^{(5,5)}$ are simultaneously active with a confidence of $\frac{3}{6}$.

A more expressive form of dependencies is made possible by the introduction of disjunctive relations, noted $\vee$. Let us consider temporal configuration of states described in Fig 5. Using conjunctive relations allows us to obtain the following dependencies: $A \wedge B \longrightarrow C \wedge D, A \wedge B \longrightarrow E, A \wedge B \longrightarrow F \wedge G \wedge H, A \wedge B \longrightarrow F \wedge G \wedge I$. This states configuration can be concisely described with a unique dependency using disjunctive relation stating:

```
IF   A AND B are active
THEN (C AND D are active) OR
     (E is active) OR
     (F and G and (H OR I) are active)
```

The corresponding dependency is noted as follows :

$$\begin{aligned} A \wedge B^{(\alpha_b,\beta_b)} \longrightarrow &(C^{(\alpha_c,\beta_c)} \wedge D^{(\alpha_d,\beta_d)}) \vee \\ &E^{(\alpha_e,\beta_e)} \vee \\ &(F^{(\alpha_f,\beta_f)} \wedge G^{(\alpha_g,\beta_g)} \wedge (H^{(\alpha_h,\beta_h)} \vee I^{(\alpha_i,\beta_i)})) \end{aligned}$$

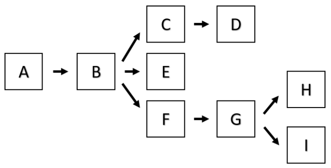


Figure 5: A temporal configuration of states. A **is followed by** B**,** B **is followed by** C **or** E **or** F **etc...**

[1] $th(lp, lc) = \dfrac{lp * lc + \sqrt{\frac{3.84}{T} lp * lc(T_{obs} - lc)(T_{obs} - lp)}}{T_{obs} * lp}$

Notice that the premise and the conclusion of this dependency are both state streams. It suffices to compute, for each dependency part, conjunctions (stream intersection), then disjunctions (stream union) to obtain a single state stream representative. This allows us to compute the confidence value following the same principle as for pairwise dependencies.

Complex Temporal Dependencies (CTD) are defined as dependencies including conjunctive and disjunctive temporal relations:

Definition 4.1. **Complex Temporal Dependency**. Let $\mathcal{S} = \{A_1, A_2, ..., A_n\}$ be a set of state streams. A Complex Temporal Dependency (CTD) over $\mathcal{S}$ is defined with:

$$D_0 \wedge D_1 \wedge ... \wedge D_k \longrightarrow D_{k+1} \wedge ... \wedge D_{k+m}$$

with D_i a state stream resulting of disjunctive and conjunctive operations using a subset $\mathcal{S}_i \subset \mathcal{S}$ such that $\forall i, j\ \mathcal{S}_i \cap \mathcal{S}_j = \emptyset$. Temporal transformations are defined with respect to a stream from D_i having a $(0, 0)$-temporal transformation.

4.3 Sub-dependencies and correspondence relationship

Conjunctive relations make it possible to obtain dependencies between multiple states reporting on "large" significant temporal correlations. As a consequence, a same temporal phenomenon involving n states can be described entirely with a single dependency or partially with multiple smaller dependencies, called sub-dependencies. We describe in Fig.6 three state streams A, B and C producing the following dependencies:

$$A \longrightarrow B^{(2,2)}, conf = 0.5 \quad (3)$$

$$B \longrightarrow C^{(2,2)}, conf = 1 \quad (4)$$

$$A \longrightarrow C^{(4,4)}, conf = 1 \quad (5)$$

$$A \wedge B^{(2,2)} \longrightarrow C^{(4,4)}, conf = 1 \quad (6)$$

In this example, dependencies (3) and (4) are sub-dependencies of (6): intervals involved in (3) and (4) are included in (6). On the other hand, dependency (5) is not a sub-dependency of (6): half of intervals of A and C do not intervene in (6). The ability to detect sub-dependencies is a key feature in a dependency discovery process that can be compared to closure-checking for support-based algorithms. In addition to reducing the amount of redundant information (pattern flooding), the sub-dependency property can be useful to define pruning criteria to accelerate a discovery process. Sub-dependency checking can be done via the study of conjunction intersection via the following *correspondence* relationship:

Definition 4.2. **Correspondence relationship** Let A and B be two state streams. A and B are corresponding if

$$conf(A \longrightarrow B) \geq 1 - \epsilon \text{ and } conf(B \longrightarrow A) \geq 1 - \epsilon$$

where ϵ is a relaxation parameter such that $\epsilon \ll 1$.

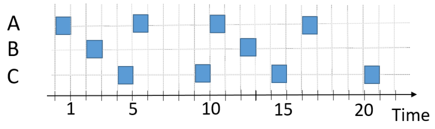


Figure 6: Dependencies $A \longrightarrow B^{(2,2)}$ and $B \longrightarrow C^{(2,2)}$ are included in $A \wedge B^{(2,2)} \longrightarrow C^{(4,4)}$. $A \longrightarrow C^{(4,4)}$ is not.

Correspondence between A and B permits to assess whether two state streams are exclusively co-occurring: A occur "almost" always with B and inversely. The correspondence checking is referred to as a boolean function $Corr\ :\ A, B, \epsilon\ \longrightarrow\ \mathbb{B}$. The ϵ parameter allows an error rate in the correspondence checking to tackle noisy or temporally variable relations. This parameter can be defined with respect to the statistical threshold on confidence measure. It corresponds to the minimum intersection length that can be considered as statistically significant. The loss of a smaller amount w.r.t the maximal confidence value (=1) can be considered as non-significant.

The sub-dependency checking between two dependencies D_1 and D_2 comes to evaluate if the representative conjunctions of both dependencies are correspondent. In the higher example, $A \wedge B^{(2,2)}$ and $A \wedge B^{(2,2)} \wedge C^{(4,4)}$ are correspondent with maximal confidence values:

$$conf(A \wedge B^{(2,2)} \longrightarrow A \wedge B^{(2,2)} \wedge C^{(4,4)}) = 1$$

$$conf(A \wedge B^{(2,2)} \wedge C^{(4,4)} \longrightarrow A \wedge C^{(2,2)}) = 1$$

On the other hand, $A \wedge C^{(4,4)}$ and $A \wedge B^{(2,2)} \wedge C^{(4,4)}$ are not correspondent:

$$conf(A \wedge C^{(4,4)} \longrightarrow A \wedge B^{(2,2)} \wedge C^{(4,4)}) = 0.5$$

$$conf(A \wedge B^{(2,2)} \wedge C^{(4,4)} \longrightarrow A \wedge C^{(4,4)}) = 1$$

In these examples, the temporal reference stream is A (all temporal transformation are expressed w.r.t to intervals of A) for both dependencies. Otherwise, a temporal transformation and an intersection are applied. For example, a $(2, 2)$ transformation is applied to $B \wedge C^{(2,2)}$ as B is shifted with a $(2, 2)$-transformation in $A \wedge B^{(2,2)} \longrightarrow C^{(4,4)}$. The sub-dependency checking is done w.r.t the intersection of this transformed stream and A (in this case: $A \wedge B^{(2,2)} \wedge C^{(2,2)}$ and $A \cap (B \wedge C^{(2,2)})^{(2,2)}$).

Definition 4.3. **Sub-dependency** Let $\mathcal{S} = \{A_1, A_2, ..., A_n\}$ be a set of state streams, R_1 a CTD with streams in $\mathcal{S}_1 \subset \mathcal{S}$ and R_2 a CTD with streams in $\mathcal{S}_2 \subset \mathcal{S}$. Then A_{R_2} A_{R_1} are respectively the temporal reference of R_1 and R_2. A_{R_2} have a (α, β)-transformation in R_1. R_2 is a sub-dependency of R_1 if $\mathcal{S}_2 \subseteq \mathcal{S}_1$ and :

$$Corr(R_1, A_{R_1} \cap R_2{}^{(\alpha,\beta)}, \epsilon) = True$$

5 PROBLEM DEFINITION

The proposed dependency model defines a search space that can be characterized in spatial and temporal dimensions. The temporal dimension contains all possible temporal transformations (α, β). Its total size for a discrete observation duration $\mathcal{T}$ is $|\mathcal{T}|^2$ as $\alpha \in \mathcal{T}$ and $\beta \in \mathcal{T}$. The spatial dimension includes all combination between state streams and dependencies operators (inference, conjunctive and disjunctive relations) in addition to the temporal dimension for all streams. This defines a very large search space even for a small set of state streams. If only conjunctive relations are taken into account, the search space size is given by:

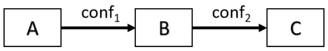
$$\sum_{k=2}^{n} \frac{n!}{(n-k)!} * (k-1) * |\mathcal{T}|^{2k}$$

with n the number of input streams. The number of conjunctive relations without temporal transformations is given by the number of arrangements of 2 to n elements from a set of n state streams. The factor $(k-1)$ stands for the possible inference operator position within a dependency. For example, with A, B and C, one can obtain $A \wedge B \longrightarrow C$ or $A \longrightarrow B \wedge C$ which are two different dependencies. Moreover, for each state in a conjunctive dependency, there are $|\mathcal{T}|^2$ possible streams to be taken into account. This search space size shows that performing a naive exploration is not affordable. Besides, this number is a lower bound of the total search space size as it does not take into account all possible disjunctive relations.

In order to perform a feasible CTD discovery, we assume that is not useful to look for dependencies with large time gaps between states (similar to a window size constraint in [7]). Therefore, we limit the allowed time lag to an interval $\Delta = [min, max]$ such that $\alpha, \beta \in \Delta$ defining a quadratic search space of size $|\Delta|^2$ for each stream. In addition to this temporal constraint, we limit our search space to dependencies with the following form:

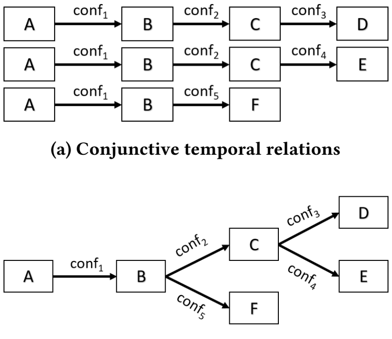
$$S_0 \wedge S_1 \wedge S_2 ... \longrightarrow S_k$$

We consider that such dependencies provide sufficient insights to describe temporal phenomena. For example, for a state succession A, B then C, obtaining $A \longrightarrow B, conf_1$ and $A \wedge B \longrightarrow C, conf_2$ permit to characterize this temporal phenomenon at each of its steps is associated with a corresponding confidence measure. This kind of characterization can be represented as a directed acyclic graph:



In this example, node A corresponds to state stream A, B to conjunction $A \wedge B$ and C to $A \wedge B \wedge C$. Edge corresponds to confidence measure: $conf_1 = conf(A \longrightarrow B)$ and $conf_2 = conf(A \wedge B \longrightarrow C)$. This representation permits to obtain a condensed representation of a given temporal phenomenon including temporal transformations.

Moreover, we assume that statistically valid disjunctive relations can be obtained only from statistically valid conjunctions. In other words, if A is temporally correlated with $B \vee C$, A is correlated with B and A is correlated with C. For example, in Fig. 7.a one can notice that the conjunctive relation $A \longrightarrow B$ is common to the 3 temporal relations. The idea is to perform of a factorization of dependencies common conjunctive relations using the correspondence relationship. The resulting disjunctive dependencies can be represented as a tree. With the former example, the corresponding tree representation is depicted in Fig. 7b. Notice that all information provided by the three conjunctive relations are kept by this disjunctive tree representation. Therefore, this paper tackles the following problem: Given a set $\mathcal{S}$ of state streams and a temporal constraint on time lags $\Delta = [min, max]$, extract all conjunctive Complex Temporal Dependencies of the form $S_0 \wedge S_1 \wedge S_2 ... \longrightarrow S_k$ and build disjunctive tree representations of corresponding temporal phenomena.



(a) Conjunctive temporal relations

(b) Corresponding disjunctive form

Figure 7: 3 conjunctive temporal relations and corresponding disjunctive form

6 CTD-MINER

CTD-Miner is based on an incremental approach consisting on building dependencies with $i + 1$ conjunction from previously computed dependencies with i conjunctions (Algorithm 1).

6.1 CTD incremental construction

The incremental construction of conjunctive relations starts with considering all streams in $\mathcal{S}$ as premise candidates (line 2). In the main loop (line 3 to 21), each pair of premise candidate p and stream $s \in \mathcal{S}$ are tested via a time lag discovery algorithm (line 13). Notice that a state label is allowed to appear only once in a dependency via the non-cyclic condition in line 12 (the conclusion state label must not appear in the premise). Results given by significant time lag discovery are used to extend p in *ExtendDependency* (line 16) that creates a new premise candidate $p \wedge r$. In order to be able to reconstruct the graph representation, the extension of dependency stores previous confidence values and temporal transformations in addition to conjunctive streams. If no results are given by the time lag discovery algorithm, p is added to the results set: p is no longer extendable. New premises at a given iteration are considered as premise candidates for the next iteration (line 21). The loop ends when no premise candidates are left. The non-cyclic condition (line 12) guarantees the termination of the main loop.

The disjunctive relations constructions constitute the final step of CTD-Miner. The results obtained after the main loop are composed of conjunctive relations. We recall that conjunctions resulting at each extension step are stored. Disjunctive dependencies are built using the correspondence relationship if two dependencies D_1 and D_2 share their k first conjunctive relation labels and the resulting conjunction are correspondent their are merged. Finally, the result set $\mathcal{R}$ is returned (line 24).

6.2 Correspondence relationship based pruning

The above steps constitute the baseline version of CTD-Miner that implements the incremental construction of conjunctive relations. The resulting dependencies may contain sub-dependencies that are considered as redundant information. For example, let us consider a temporal phenomenon, called A, constituted of n temporally consecutive states $A_1 \longrightarrow A_2 ... \longrightarrow A_n$. The results set will contain dependencies describing all or a subdivision of this phenomenon,

Algorithm 1: CTD-Miner

```
Input: $\mathcal{S}$ : a set of state streams
       $\mathcal{T}$ : observation duration
       $\Delta = [min, max]$ : temporal constraint on time lags
Output: $\mathcal{R}$ : set of CTDs
1  $\mathcal{R} \leftarrow \emptyset$
2  premises $\leftarrow \mathcal{S}$
3  while $|premises| > 0$ do
4      $newPremises \leftarrow \emptyset$
5      for $p \in premises$ do
6          $extended \leftarrow False$
7          $process \leftarrow True$
8          if $|p| > 1$ then
9              $process \leftarrow$ PrePruning($p, newPremises \cup \mathcal{R}$)
10         if $process$ then
11             for $s \in \mathcal{S}$ do
12                 if $s.label \notin p.labels$ then
13                     results $\leftarrow$ TimeLagDiscovery($p$, $s$, $\Delta$, $\mathcal{T}$)
14                     for $r \in results$ do
15                         $extended \leftarrow True$
16                         $ext \leftarrow$ ExtendDependency($p, r$)
17                         Add $ext$ to $newPremises$
18             if $|p| > 1$ and not $extended$ then
19                 Add $p$ to $\mathcal{R}$
20     $newPremises \leftarrow$ MergeDependencies($newPremises$)
21     $premises \leftarrow newPremises$
22 $\mathcal{R} \leftarrow$ PostProcessing($\mathcal{R}$)
23 $\mathcal{R} \leftarrow$ BuildDisjunctions($\mathcal{R}$)
24 return $\mathcal{R}$
```

$A_{n-1} \longrightarrow A_n$, $A_{n-2} \wedge A_{n-1} \longrightarrow A_n$ ... $A_1 \wedge A_2 ... \wedge A_{n-1} \longrightarrow A_n$. Besides, for each of the previous dependencies, the conjunction relations order can differ while maintaining temporal information: $A_1 \wedge A_3^{(\alpha_3, \beta_3)} \rightarrow A_2^{(\alpha_2, \beta_2)}$ and $A_1 \wedge A_2^{(\alpha_2, \beta_2)} \rightarrow A_3^{(\alpha_3, \beta_3)}$ corresponds to the same temporal succession of states. Therefore, the total number of resulting conjunctive dependencies describing this n-states temporal phenomenon is given by:

$$|\mathcal{R}|_A = \sum_{k=2}^{n}(k-1)! = \sum_{k=1}^{n-1} k!$$

In order to keep the number of solutions moderate, CTD-Miner benefits from a sub-dependency checking at two levels. First, as a pre-pruning step (line 9) via the PrePruning function that performs a sub-dependency checking of a premise p w.r.t dependencies in $newPremises$ and $\mathcal{R}$. If p is a sub-dependency of another, it is pruned: *PrePruning* returns false and p is not processed nor added to the result set. Second, to ensure that no sub-dependencies are left in the final results, a post processing step (line 22) is performed.

Another way to benefit from the correspondence relationship is that of merging dependencies (line 20). This operation consists of extending dependencies with other dependencies rather than a single stream. For example, $A \longrightarrow B^{(\alpha_b, \beta_b)}$ and $B \longrightarrow C$ can be merged if $A \wedge B^{(\alpha_b, \beta_b)}$ and $A \cap (B \wedge C)^{(\alpha_b, \beta_b)}$ are correspondent.

6.3 Time lag discovery

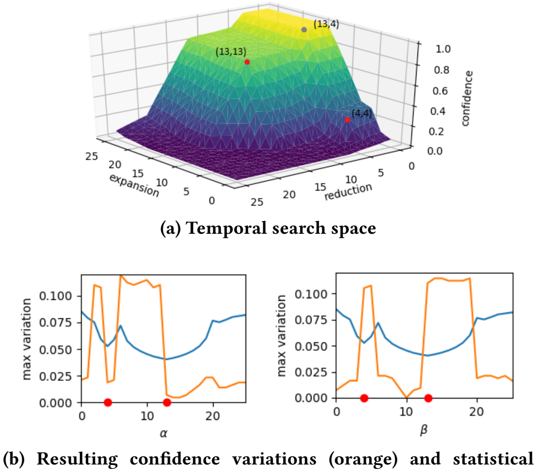
The discovery of temporally quantitative correlation between two state streams is a core operation in CTD-Miner. This operation is executed at a high frequency during the discovery process. As a consequence, its time complexity influences significantly the performance of CTD-Miner.

In order to discover such correlations, we designed the ITLD (Interval Time Lag Discovery) heuristic algorithm that we describe briefly in this section. It performs a linear significant time lag discovery w.r.t Δ. The main idea of this algorithm is to detect statistical significant confidence gains and losses. More precisely, given a pair of state stream A and B, for each i $\alpha \in \Delta$, ITLD computes the following confidence variations:

$$gain = conf(A \longrightarrow B^{(\alpha, \alpha)}) - conf(A \longrightarrow B^{(\alpha+1, \alpha)})$$

$$loss = conf(A \longrightarrow B^{(\alpha, \alpha)}) - conf(A \longrightarrow B^{(\alpha, \alpha+1)})$$

The confidence gains and losses are calculated w.r.t a non deformed conclusion streams i.e with (α, α)-transformations. This permits to guarantee that intervals of the conclusion B cannot be merged or canceled due to an important deformation induced by temporal transformations, which constitute a loss of transition information. These confidence variations are qualified as elementary as they are obtained by adding a unit to whether the expansion (for gains) or the reductions (for losses). As a consequence, the conclusion gained/lost length corresponds to number of intervals of B. The statistical assessment of these elementary confidence variations is done with the validity threshold $th(len(A \cap B^{(\alpha, \alpha)}, \#B)$ is it aims to evaluate whether the confidence variation is statistically independent w.r.t the gained/lost conclusion length. We describe in Fig. 8 the entire



(a) Temporal search space

(b) Resulting confidence variations (orange) and statistical thresholds (blue). Red points refer to significant expansion and reduction values

Figure 8: Temporal search space (a) and resulting confidence variations for a pair of dependencies with two temporal relations $(13, 13)$ **and** $(4, 4)$**.** $\Delta = [0, 25]$

search space (a) and the corresponding confidence gains, losses and thresholds.

7 EXPERIMENTS

This section presents the experimental results that illustrate the efficiency of CTD-Miner. We experimented our approach using synthetic data provided by a motion simulation tool that makes it possible to obtain data corresponding to different scenarios. We also study the behaviour of CTD-Miner with a real world motion data set generated from a sensor system composed of outdoor video cameras and using real time video processing. Algorithms were implemented in Python[2] and tested on a Core i7 2.1Ghz with 8GB memory running Windows 10.

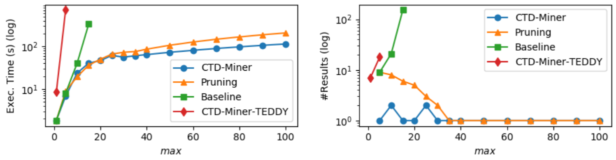
7.1 Simulated Data

We developed a motion simulation tool[2] in order to obtain data sets using custom scenarios (trajectories, activity density, speed of moving objects, temporal variability). This controlled testbed generates data sets with a known ground truth permitting to evaluate the accuracy of discovered temporal dependencies in a multitude of configurations.

Table 1 describes datasets obtained from the simulation tool. We defined a linear trajectory with 10 equidistant sensors and ran 11 simulations varying the number of occurrences for the same duration $\mathcal{T} = 10000$ and object speed (cf Table 1). Each dataset contains 10 streams and the typical time lag between successive state is $\approx (4, 4)$. The number of intervals increases with the number of event occurrences for sparse data (100 to 5000 occurrences) and decreases for high density event occurrences due to intervals overlap. The streams active length always increases when the event occurrences increases and have a density ($\frac{len(stream)}{T_{obs}}$) ranging from 1.7% to 78.5%. It is to notice that the expected result from these datasets is the trajectory description including 10 states with the corresponding (α, β)-transformations (represented as a 10 nodes graph).

Hereafter, we evaluate our approach with respect to size of Δ, density of streams intervals and scalability. We also examined the impact of sub-dependencies pruning, dependencies merging and the time lag discovery algorithm on CTD-Miner's efficiency. To this end, we consider the following CTD-Miner configurations: **Baseline**: consists of the simple incremental construction of CTD as described in section 6.1; **Pruning**: line 20 is removed from Algorithm 1 and only the sub-dependencies pruning is performed ; **CTD-Miner**: CTD-Miner as described in Algorithm 1. ITLD was used for the three previous algorithm. The fourth CTD-Miner configuration is **CTD-Miner-TEDDY** with TEDDY [11] instead of ITLD. For the following tests, we limited the execution of the CTD discovery algorithms to 20 minutes and used the statistical threshold for the correspondence relationship.

7.1.1 Influence of the size of Δ. Fig.9 reports the running time and number of conjunctive relations (before disjunctions construction) for the 1000 occurrences dataset when $\Delta = [0, max]$ varies.



(a) Running time

(b) Number of results for different values of *max*

Figure 9: Running time and number of results of CTD-Miner, Pruning, Baseline, CTD-Miner-TEDDY w.r.t to *max*

CTD-Miner-TEDDY and Baseline reached the 20 minutes running time limit for respectively $max = 10$ and $max = 20$. In both cases, the number of premise candidates, and by extension the number of time lag discovery executions, increases significantly. This can be noticed with the number of the given conjunctive relations. TEDDY outputs a non-negligible amount of false positives w.r.t the simulation scenario. For example, with $\Delta = [0, 1]$ TEDDY obtained 7 dependencies when none is expected. Candidate subdependencies increases significantly for the Baseline as the number of given conjunctive dependencies shows.

CTD-Miner and Pruning were capable of completing the discovery process for all values of $max \in [0, 100]$ with an advantage for CTD-Miner. Results for Pruning corresponds to dependencies with time lags in a given $\Delta = [0, max]$ while CTD-Miner returns a unique dependency corresponding to the expected 10 conjunctions dependency for $max \geq 5$ (except for 10 and 25 where Δ did not permit to fully capture the intersection for a temporal relation with $\alpha = max + 1$ and $\beta = max + 1$, causing the non-verification of the correspondence relationship). It is to notice that Pruning obtains the same result for $max \geq 35$. Thus, thanks to the sub-dependency pruning and the dependency merging CTD-Miner scales well with respect to Δ.

7.1.2 Influence of density of streams. As described in Table 1, activity density (number of occurrence) comes generally with an increase of length and intervals number. Therefore, a complex temporal dependency discovery algorithm must be capable of scaling with respect to intervals numbers and be able to detect precisely temporal phenomena in both sparse and dense state streams.

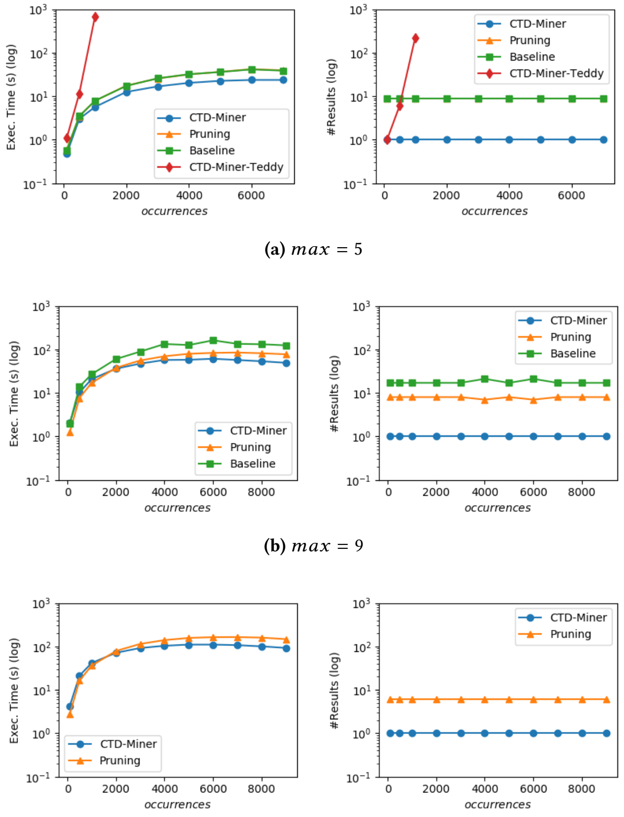
Fig.10 reports the empirical study of CTD-Miner with respect to streams density. The constant number of conjunctive results given by CTD-Miner with the ITLD (CTD-Miner, Pruning and Baseline) for each value of max shows that ITLD is robust to density of streams and allows CTD-Miner to scale well with respect to intervals number for these datasets. This is not the case for CTD-Miner-Teddy due to its time lag discovery algorithm (that 20 minutes limit was reached for 2000 occurrence with $max = 5$).

7.1.3 Scalability. One important aspect of CTD discovery is that of the ability of processing large numbers of streams. We generated a data set of 90 streams following the same principle as data sets described in Table 1 with $\mathcal{T} = 10000$ time units, 1000 occurrences and a time lag of $\approx (6, 6)$ between successive states. Fig.11 shows that CTD-Miner was able to process 90 streams (containing 63990

[2] https://github.com/AElOuassouli/Quantitaive-Interval-Stream-Mining

Table 1: 11 simulated data sets with increasing occurrences number (*Occ*). *#Int*: average number of intervals per stream, *Den*: average density % of T_{obs}

Occ	#Int	Den
100	95	1.7
500	452	8.7
1000	798	16.4
2000	1275	30.3
3000	1509	41.5
4000	1606	50.5
5000	1616	58.6
6000	1570	64.7
7000	1493	69.9
8000	1386	74.6
9000	1263	78.5



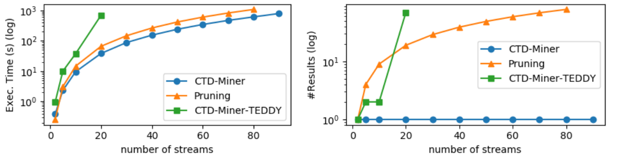
(a) $max = 5$

(b) $max = 9$

(c) $max = 15$

Figure 10: Running time and Number of conjunctive results with respect to occurrence number with different values of *max*

intervals) within the 20 minutes time limit. CTD-Miner returned a unique dependency describing the entire trajectory, with all available streams, for all input sizes thanks to its merging procedure contrary to Pruning which is limited by the temporal constraint $\Delta = [0, 9]$ and returns an increasing number of pairwise dependencies. Notice that for this particular Δ, the performances of Pruning are equivalent to Baseline as the sub-dependency pruning is not performed. CTD-Miner-TEDDY reaches the 20 minutes for 30 streams and returns the trajectory description in addition to a large number of dependencies that are considered as false positives w.r.t to the simulation scenario.



(a) Running Time

(b) Number of conjunctive relations

Figure 11: Running time and number of conjunctive relations w.r.t to number of streams. $\Delta = [0, 9]$

7.2 Real life motion data

We describe in Fig. 12 and Fig. 13 a sensor system composed of 4 outdoor cameras situated in an office area. These cameras are capturing motion using real time video processing. Starting from images taken by these cameras, we defined 10 "virtual" motion sensors (displayed with red polygons) corresponding to physical regions and labelled as 1-1, 1-2, ..., 4-3. Each virtual motion sensor produces a sequence of time point events of two types: *"B: Motion Begin"* and *"E: Motion End"*. We defined for each area X an

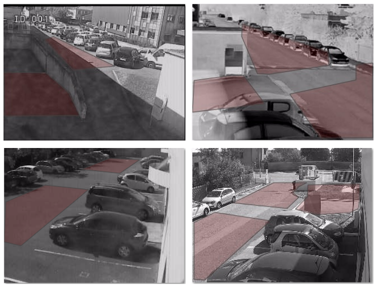


Figure 12: Four outdoor camera views. Red contours describe motion analysis areas.

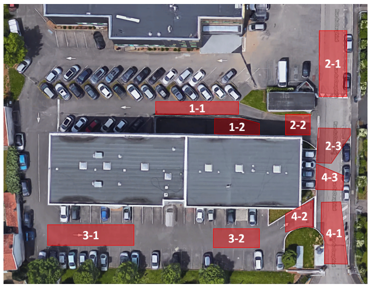


Figure 13: Position of motion analysis areas (aerial view)

Table 2: Dataset corresponding to the experiment described in Fig.13

Name	#Int	Len
1-1	1403	4303
1-2	851	3257

Name	#Int	Len
2-1	8640	47881
2-2	2099	4947
2-3	8548	26847

Name	#Int	Len
3-1	3909	14644
3-2	3699	13423

Name	#Int	Len
4-1	9686	30257
4-2	4578	13397
4-3	9825	21273

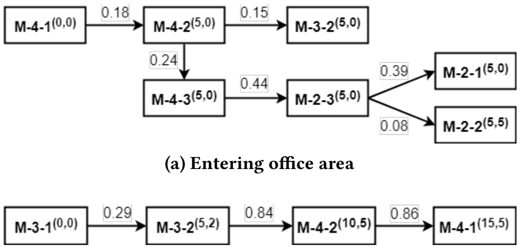
environment state *Motion in area X* noted **M-X** and defined with **M-X**(t,X) ::= $last(t, X).v == B$. The 10 resulting state streams are described in Table 2. This data set describes motion activity in the office area during 18 working days between 6 am and 8 pm. First observations showed that it contains a significant amount of noise (e.g detection of shadows, sudden luminosity changes) and several omissions were observed (e.g when a car passes through an analysis zone with a great speed). Moreover, the resulting streams are extremely sparse and low statistical thresholds impacts the correspondence relationship. As a consequence, we used $\epsilon = 0.2$ as relaxation parameter. The low statistical thresholds impacts to some extent the precision of ITLD: thresholds were in some cases lowers than significant confidence variations. In our experiments we used the statistical thresholds [1] with a significance level of 0.005 (critical value of $\chi^2_{0.005} = 7.88$ instead of $\chi^2_{0.05} = 3.84$) to obtain more precise results.

The execution of CTD-Miner with $\Delta = [0, 5]$ was completed in 300 seconds and provided 12 disjunctive relations. Figure 14 provides two examples that illustrate two behaviours: entering office area (a) and leaving office area (b). What is to notice is that the first dependency was built using the incremental construction of CTDs all time lags are included in Δ and the second using dependencies merge. Moreover none of the given dependencies provides false information (e.g relation between *2-1* and *3-1*). We emphasize the fact that the discovery process was completed in a negligible amount of time in comparison with the observation duration.

8 CONCLUSION AND FUTURE WORK

We proposed an approach to discover complex temporal dependencies (CTD) starting from interval based state streams. These streams are built through temporal abstractions on heterogeneous data. Each temporal abstraction (state) is defined using a predicate defining a state of interest for the application domain. We introduced the Complex Temporal Dependencies model that models temporal relations between state streams and their typical time delays using conjunctive and disjunctive relations. We also proposed CTD-Miner, an incremental algorithm that is devised to CTD discovery. To validate our model and the discovery process, we conducted several experiments on both simulated and real motion sensor data. From these results, we conclude that CTD-Miner speeds up the exploration process using its linear time delay discovery, its sub-dependency pruning and dependencies merge. The encouraging results given for the real data set show that it is possible to integrate video analysis methods in a data analysis process which opens perspectives for a wide range of application scenarios. For example, in a commercial context, our approach may permit to investigate temporal dependencies between user-given streams as commercial results or client satisfaction rate, sensory data streams as motion, people counting, and video processing-based streams as objects classification (adults, stroller).

In a future work, we intend to design an on-line version of CTD-Miner in order to obtain up-to-date models of temporal phenomena occurring within a sensed environment. This would permit to investigate the ability to detect behavior drifts (emergence of new dependencies) in addition to the forecasting capabilities of our intersection-based dependency model.



(a) Entering office area

(b) Leaving office area

Figure 14: Example given by CTD-Miner on real world data generated by environment depicted in Figure 13

REFERENCES

[1] J. F. Allen and P. J. Hayes. A common-sense theory of time. In *IJCAI*, pages 528–531. Morgan Kaufmann, 1985.

[2] G. Casas-Garriga. Discovering unbounded episodes in sequential data. In *PKDD*, volume 2838 of *Lecture Notes in Computer Science*, pages 83–94. Springer, 2003.

[3] T. Guyet and R. Quiniou. Extracting temporal patterns from interval-based sequences. In *IJCAI*, pages 1306–1311. IJCAI/AAAI, 2011.

[4] M. Hassani, Y. Lu, J. Wischnewsky, and T. Seidl. A geometric approach for mining sequential patterns in interval-based data streams. In *FUZZ-IEEE*, pages 2128–2135. IEEE, 2016.

[5] F. Höppner and S. Peter. Temporal interval pattern languages to characterize time flow. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 4(3):196–212, 2014.

[6] T. Li and S. Ma. Mining temporal patterns without predefined time windows. In *ICDM*, pages 451–454. IEEE Computer Society, 2004.

[7] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.*, 1(3):259–289, 1997.

[8] F. Mörchen. Algorithms for time series knowledge mining. In *KDD*, pages 668–673. ACM, 2006.

[9] R. Moskovitch and Y. Shahar. Fast time intervals mining using the transitivity of temporal relations. *Knowl. Inf. Syst.*, 42(1):21–48, 2015.

[10] F. Nakagaito, T. Ozaki, and T. Ohkawa. Discovery of quantitative sequential patterns from event sequences. In *ICDM Workshops*, pages 31–36. IEEE Computer Society, 2009.

[11] M. Plantevit, C. Robardet, and V. Scuturici. Graph dependency construction based on interval-event dependencies detection in data streams. *Intell. Data Anal.*, 20(2):223–256, 2016.

[12] G. Ruan, H. Zhang, and B. Plale. Parallel and quantitative sequential pattern mining for large-scale interval-based temporal data. In *BigData*, pages 32–39. IEEE Computer Society, 2014.

[13] L. Tang, T. Li, and L. Shwartz. Discovering lag intervals for temporal dependencies. In *KDD*, pages 633–641. ACM, 2012.

[14] W. Wang, C. Zeng, and T. Li. Discovering multiple time lags of temporal dependencies from fluctuating events. In *APWeb/WAIM (2)*, volume 10988 of *Lecture Notes in Computer Science*, pages 121–137. Springer, 2018.

An Effective Algorithm for Learning Single Occurrence Regular Expressions with Interleaving

Yeting Li
State Key Laboratory of Computer Science, Institute of Software Chinese Academy of Sciences
University of Chinese Academy of Sciences
Beijing, China
liyt@ios.ac.cn

Haiming Chen
State Key Laboratory of Computer Science, Institute of Software Chinese Academy of Sciences
Beijing, China
chm@ios.ac.cn

Xiaolan Zhang
State Key Laboratory of Computer Science, Institute of Software Chinese Academy of Sciences
University of Chinese Academy of Sciences
Beijing, China
Zhangxl@ios.ac.cn

Lingqi Zhang
Beijing University of Technology
Beijing, China
zhanglingqisteve@gmail.com

ABSTRACT

The advantages offered by the presence of a schema are numerous. However, many XML documents in practice are not accompanied by a (valid) schema, making schema inference an attractive research problem. The fundamental task in XML schema learning is inferring restricted subclasses of regular expressions. Most previous work either lacks support for interleaving or only has limited support for interleaving. In this paper, we first propose a new subclass *Single Occurrence Regular Expressions with Interleaving* (SOIRE), which has unrestricted support for interleaving. Then, based on *single occurrence automaton* and *maximum independent set*, we propose an algorithm *i*SOIRE to infer SOIREs. Finally, we further conduct a series of experiments on real datasets to evaluate the effectiveness of our work, comparing with both ongoing learning algorithms in academia and industrial tools in real-world. The results reveal the practicability of SOIRE and the effectiveness of *i*SOIRE, showing the high preciseness and conciseness of our work.

CCS CONCEPTS

• **Information systems** → **Data management systems**; **Web data description languages**.

KEYWORDS

XML, schema inference, learning expressions, interleaving

ACM Reference Format:
Yeting Li, Haiming Chen, Xiaolan Zhang, and Lingqi Zhang. 2019. An Effective Algorithm for Learning Single Occurrence Regular Expressions with Interleaving. In *IDEAS 2019: 23rd International Database Engineering & Applications Symposium, June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

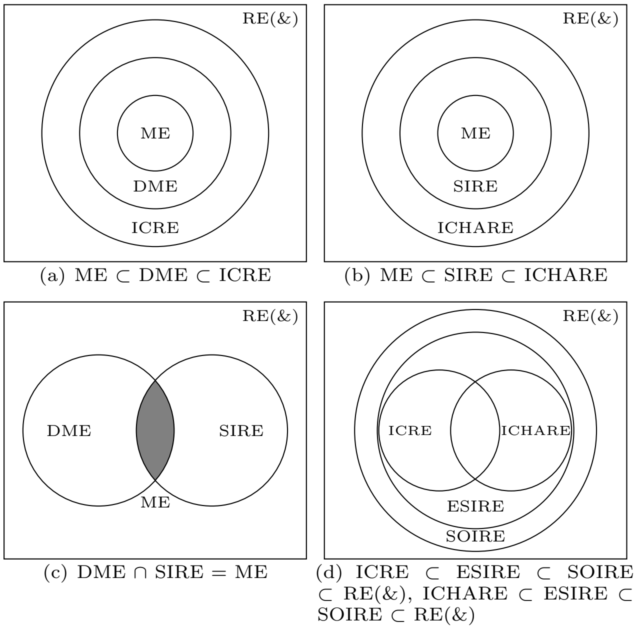
XML schemas have always played a crucial role in XML management. The presence of a schema for XML documents has many advantages, such as for query processing and optimization, development of database applications, data integration and exchange [15, 18, 34, 42]. However, many XML documents in practice are not accompanied by a (valid) schema [3, 6, 25, 36, 37, 41], making schema inference an attractive research problem [2, 5, 7, 13, 17, 22, 30, 32, 43]. Studying schema inference also has several practical motivations. Schema inference techniques may be extended to schema repairing techniques [25]. Besides, schema inference is also useful in situations where a schema is already available, such as in schema cleaning and dealing with noise [7].

The content models of XML schemas are defined by regular expressions, and previous research has shown that the essential task in schema learning is inferring regular expressions from a set of given samples [2, 5, 7, 9, 13, 17, 22, 30, 32, 43]. In fact, in some cases these learned regular expressions can directly be used as parts of the schema, and in other cases the inference of regular expressions is the most important component of the schema inference. Therefore, research on schema learning has focused on inferring regular expressions from a set of given samples.

We focus on learning regular expressions with *interleaving* (*shuffle*), denoted by RE(&). Since RE(&) are widely used in various areas of computer science [4], including XML database systems [14, 19, 34], complex event processing [33], system verification [10, 21, 23], plan recognition [26] and natural language processing [27, 39].

IDEAS 2019, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-9999-9/18/06. . . $15.00
https://doi.org/10.1145/1122445.1122456



(a) ME ⊂ DME ⊂ ICRE (b) ME ⊂ SIRE ⊂ ICHARE

(c) DME ∩ SIRE = ME (d) ICRE ⊂ ESIRE ⊂ SOIRE ⊂ RE(&), ICHARE ⊂ ESIRE ⊂ SOIRE ⊂ RE(&)

Figure 1: Relationships among ME, DME, SIRE, ICRE, ICHARE, ESIRE, SOIRE and RE(&).

Inference of regular expressions from a set of given samples belongs to the problem of language learning. Gold proposed a classical language learning model (*learning in the limit or explanatory learning*) and pointed out that the class of regular expressions could not be identifiable from positive samples only [24]. This means that no matter how many positive samples from the target language (i.e., the language to be learned) are provided, no algorithm can infer every target regular expression. Hence, researchers have turned to study subclasses of regular expressions [2, 5, 7, 9, 13, 17, 22, 30, 32, 38, 43].

Most existing subclasses of regular expressions for XML are defined on standard regular expressions, e.g., [5–7, 16, 35] which were analyzed together in [28, 31]. For single occurrence regular expressions (SOREs), in which each symbol occurs at most once and its subclass chain regular expressions (CHAREs), Bex et al. proposed two inference algorithms *RWR* and *CRX* [7, 8]. Freydenberger and Kötzing [17] proposed more efficient algorithms *Soa2Sore* and *Soa2Chare* for the above mentioned SOREs and CHAREs. Bex et al. [5] also studied learning algorithms, based on the Hidden Markov Model, for the subclass of regular expressions (k-OREs) in which each alphabet symbol occurs at most k times. Notice that none of the above subclasses support an important feature in XML, i.e., the interleaving.

There may be no order constraint among siblings in datacentric applications [1]. In such cases the interleaving is necessary. Here we list the more recent efforts on RE(&) inference (see [13, 30, 32, 40, 43]). The aim of these approaches is to infer restricted subclasses of single occurrence RE(&), in which each symbol occurs at most once, starting from a positive set of words. Ciucanu and Staworko proposed two subclasses disjunctive multiplicity expression (DME) and disjunction-free multiplicity expression (ME) [11, 13] which support unordered concatenation, a weaker form of interleaving. The concatenation operator is disallowed in both formalisms and ME even uses no disjunction operator. For example, $r_1 = (a|b^+)\&c$ is a DME and $r_2 = a\&b^*\&c^?$ is an ME. But $r_3 = (a^+b^?)\&c^*$ and $r_4 = a^*((b^*|c)\&d^*)$ do not satisfy both formalisms. The inference algorithm based on *maximum clique* for DME was given in [13]. Li et al. provided an algorithm to learn DMEs from both positive and negative examples based on genetic algorithms and simplified candidate regions (SCRs) [29]. When there is no order constraint among siblings, the relative orders within siblings are still important. Peng and Chen [40] proposed a subclass SIRE using the grammar: $S ::= T\&S|T$, $T ::= \varepsilon|a|a^*|TT$. But it does not support the union operator. For example, r_2 and r_3 are SIREs but r_1 and r_4 are not. Besides, they presented an approximate algorithm to infer SIREs [40]. Li et al. [30] proposed a subclass ICRE using the grammar:

$$\begin{aligned} E &:= F_1^{p_1} \cdot \ldots \cdot F_n^{p_n}, && (n \geq 0, p_i \in \{?, 1\}),\\ F_i &:= D_1\& \ldots \&D_k, && (i \in [1,n], k \geq 1),\\ D_j &:= a_1^{mul_1}| \ldots |a_m^{mul_m}, && (j \in [1,k], m \geq 1), \end{aligned}$$

where $mul_o \in \{1, ?, *, +\}$ and $a_o \in \Sigma$ for $o \in [1,m]$. For example, r_1, r_2 and r_4 are ICREs but r_3 is not. Besides, they presented an approximate algorithm to infer ICREs [30]. Zhang et al. [43] proposed a subclass called ICHARE considering interleaving. The inference algorithm is based on SOA and maximum independent set (MIS). However, components of interleaving are restricted to the *extended strings* (ES) defined in [43]. For example, r_2 and r_3 are ICHAREs but r_1, r_4 and $r_5 = a^?((b^+|c)d^*\&ef^?)$ are not. Li et al. [32] proposed a practical subclass called ESIRE and designed an inference algorithm GenESIRE to infer ESIREs. For example, r_1, r_2, r_3, r_4 and r_5 are ESIREs, but $r_6 = a^*b^?(fm^?\&c^?d|e(n|l)^?g\&h^?)(j^+|k)^?$ is not. All of the above subclasses are restricted subclasses of single occurrence RE(&). As shown above, the support for interleaving in existing work is very limited.

In this paper, based on the analysis of large-scale real data, we propose a new subclass of RE(&), i.e., single occurrence RE(&), called SOIRE. The relationships among ME, DME, SIRE, ICRE, ICHARE, ESIRE, SOIRE and RE(&) are shown in Figure 1. Among them, ME ⊂ DME ⊂ ICRE, ME ⊂ SIRE ⊂ ICHARE, DME ∩ SIRE = ME, ICRE ⊂ ESIRE ⊂ SOIRE ⊂ RE(&) and ICHARE ⊂ ESIRE ⊂ SOIRE ⊂ RE(&). For example, all of r_1, r_2, r_3, r_4, r_5 and r_6 are SOIREs. It reveals that SOIRE is more powerful than the above subclasses since the latter are all subclasses of SOIRE, and especially SOIRE has unrestricted support for interleaving, which was never achieved by existing work. Then, we develop the corresponding learning algorithm, iSOIRE, to carry out SOIREs inference automatically. The massive experimental results demonstrate the practicality of the proposed subclass as well as the preciseness and conciseness of *i*SOIRE.

The main contributions of this paper are listed as follows.

- We propose a new subclass SOIRE of RE(&). SOIRE is more powerful than the existing subclasses and especially has unrestricted support for interleaving.
- Correspondingly, we design an inference algorithm *i*SOIRE which can learn SOIREs effectively based on single occurrence automaton (SOA) and maximum independent set (MIS).
- We conduct a series of experiments, comparing the performance of our algorithm with both ongoing learning algorithms in academia and industrial tools in real-world. The results reveal the practicability of SOIRE and the effectiveness of *i*SOIRE, showing the high preciseness and conciseness of our work.

The rest of this paper is organized as follows. Preliminaries are presented in Section 2. Section 3 provides the learning algorithm. Then a series of experiments is presented in Section 4. Finally we conclude this work in Section 5.

2 PRELIMINARIES

2.1 Definitions

Let Σ be a finite alphabet of symbols. The set of all words over Σ is denoted by Σ^*. The empty word is denoted by ε.

Definition 2.1. **Regular Expression with Interleaving.** A regular expression with interleaving over Σ is defined inductively as follows: ε or $a \in \Sigma$ is a regular expression, for regular expressions r_1 and r_2, the disjunction $r_1|r_2$, the concatenation $r_1 \cdot r_2$, the interleaving $r_1\&r_2$, or the Kleene-Star r_1^* is also a regular expression. $r^?$ and r^+ are abbreviations of $r|\varepsilon$ and $r \cdot r^*$, respectively. They are denoted as RE(&).

The size of a regular expression r, denoted by $|r|$, is the total number of symbols and operators occurred in r. The language $L(r)$ of a regular expression r is defined as follows: $L(\varnothing) = \varnothing$; $L(\varepsilon) = \{\varepsilon\}$; $L(a) = \{a\}$; $L(r_1^*) = L(r_1)^*$; $L(r_1 \cdot r_2) = L(r_1)L(r_2)$; $L(r_1|r_2) = L(r_1) \cup L(r_2)$; $L(r_1\&r_2) = L(r_1)\&L(r_2)$. Let $u = au'$ and $v = bv'$ where $a, b \in \Sigma$ and $u', v' \in \Sigma^*$, then $u\&\varepsilon = \varepsilon\&u = \{u\}$ and $u\&v = a(u'\&v) \cup b(u\&v')$. For example, $L(ab\&c) = \{cab, acb, abc\}$.

Definition 2.2. **Single Occurrence Regular Expressions with Interleaving (SOIRE).** A regular expression with interleaving is SOIRE, in which each symbol occurs at most once.

For instance, $r_1 = a^?(b^?c\&d^*(e|f)^?)$ is an SOIRE, but $r_2 = a^+b\&c^+b$ is not because b appears twice.

Definition 2.3. **Single Occurrence Automaton (SOA)** [7, 17] Let Σ be a finite alphabet. *src* and *snk* are distinct symbols that do not occur in Σ. A single occurrence automaton (short: SOA) over Σ is a finite directed graph $\mathcal{A} = (V, E)$ such that

(1) $src, snk \in V$, and $V \subseteq \Sigma \cup \{src, snk\}$;

(2) *src* has only outgoing edges, *snk* has only incoming edges and every node $v \in V$ lies on a path from *src* to *snk*.

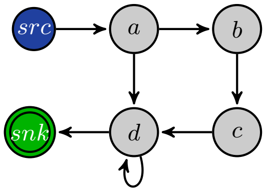


Figure 2: Example SOA $\mathcal{A}$ for $r = a(bc)^?d^+$.

For example, the SOA $\mathcal{A}$ for $r = a(bc)^?d^+$ is shown in Figure 2. A **generalized single occurrence automaton (generalized SOA)** over Σ is defined as a directed graph in which each node $v \in V \setminus \{src, snk\}$ is an SOIRE and all nodes are pairwise alphabet-disjoint SOIREs.

3 LEARNING ALGORITHM

In this section, we give the learning algorithm *i*SOIRE, which efficiently infers an SORE from a set of positive samples S. We show the major technical details of our algorithm in this section. The input and output of the algorithm *i*SOIRE is a set of given samples and an SOIRE respectively. The algorithm *i*SOIRE consists of two steps, constructing an SOA from samples, and converting the SOA into an SOIRE. Constructing an SOA from samples is introduced in Section 3.1. Converting the SOA into an SOIRE is given in Section 3.2.

```
Algorithm 1: iSOIRE
  Input: a set of positive sample $S$
  Output: an SOIRE
  Construct SOA $\mathcal{A}$ for $S$ using method 2T-INF [20];
  return Soa2Soire($S$, $\mathcal{A}$)
```

3.1 Constructing an SOA from Samples

We use method 2T-INF [20] to construct SOA $\mathcal{A}$ for S. The algorithm 2T-INF [20] used in the algorithm is proved to construct a minimal-inclusion generalization of S. Here minimal-inclusion means that there is no other *SOA* $\mathcal{A}$ such that $S \subseteq L(\mathcal{A}) \subset L(SOA(S))$.

Here we give an example to show the execution process. Let $S = \{begk, aabengk, abegjj, beg, hk, behgj, belhg, bheg, bfcmd, bfdm, afmcd, adf\}$. Using method *2T-INF*, we construct the graph *SOA(S)* shown in Figure 3.

3.2 Converting the SOA into an SOIRE

We use dot-notation to denote the application of subroutines. For a given SOA $\mathcal{A}$, we let $\mathcal{A}.src$ and $\mathcal{A}.snk$ denote the source and the sink of $\mathcal{A}$, respectively. We let V be the set of vertices and E the set of edges in $\mathcal{A}$, respectively.

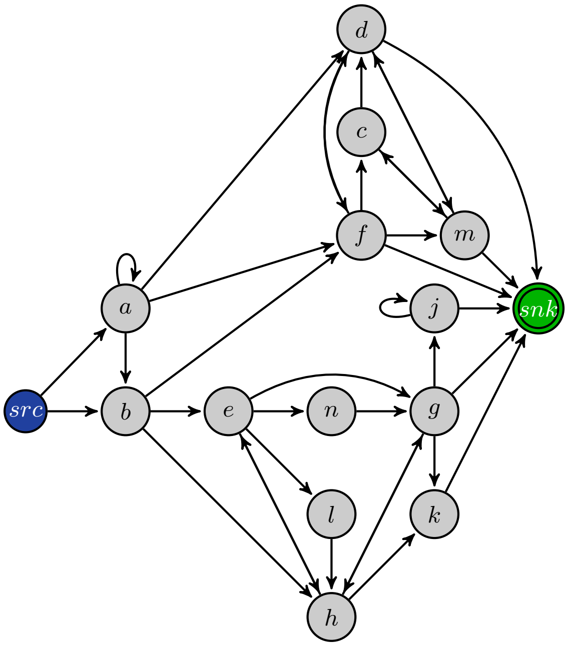


Figure 3: Constructing SOA $\mathcal{A}$ for S.

- For any vertex $v \in V$, we let $\mathcal{A}$.pred(v) denote the set of all predecessors of v in $\mathcal{A}$; similarly, $\mathcal{A}$.succ(v) denotes the set of all successors of v in $\mathcal{A}$.
- For any vertex $v \in V$, we let $\mathcal{A}$.reach(v) be the set of all vertices reachable from v.
- "first" returns all vertices v such that the only predecessor of v is the source in $\mathcal{A}$.
- "contract" on SOA $\mathcal{A}$ takes a subset U of vertices of $\mathcal{A}$ and a label δ. The procedure modifies $\mathcal{A}$ such that all vertices of U are contracted to a single vertex and labeled δ (edges are moved accordingly).
- "extract" on SOA $\mathcal{A}$ takes as argument a set of vertices U of $\mathcal{A}$; it does not modify $\mathcal{A}$, but returns a new SOA with copies of all vertices of U as well as two new vertices for source and sink; all edges between vertices of U are copied, all vertices in U having an incoming edge in $\mathcal{A}$ from outside of U have now an incoming edge from the new source, and all vertices in U having an outgoing edge in $\mathcal{A}$ to outside of U have now an outgoing edge to the new sink.
- "addEpsilon" on SOA $\mathcal{A}$ adds a new vertex labeled ε; all outgoing edges from the source to vertices that have more than one predecessor (vertices, that are not in the first-set) are redirected via this new vertex.
- "exclusive" on SOA $\mathcal{A}$ on argument v (a vertex of $\mathcal{A}$) returns the set of all vertices u such that, on any path from the source to the sink that visits u, v is necessarily visited previously. Intuitively, the exclusive set of a vertex v is the set of all vertices exclusively reachable from v, not from any other vertex incomparable to v.

Furthermore, we use the following eight subroutines or algorithms.

- "plus" on label δ returns δ^+.
- "or" on labels δ and δ' returns $\delta|\delta'$.
- "concatenate" on labels δ and δ' returns $\delta \cdot \delta'$.
- "filter" on a subset U of vertices and a set of given sample S returns a new subset S'. For string $s \in S$ each symbol of which is computed as follows: $\pi_s(U, s_i) = s_i$ if $s_i \in U$; $\pi_s(U, s_i) = \varepsilon$ otherwise. And the result is reduced by $x\varepsilon = \varepsilon x = x$. For example, let U=$\{b, c, r\}$ and $S = \{abgr, ebbdfc\}$, $S' = filter(U, S) = \{br, bbc\}$.
- "Merge" on a set of positive samples S returns an expression ζ with interleaving.
- For a set of positive sample S, we let por(S) denote the set of all partial order relations of each string in S and cs(S) denote the constraint set. The cs(S) is defined as follows. $cs(S) = \{\langle x, y\rangle | \langle x, y\rangle \in por(S)\ and\ \langle y, x\rangle \in por(S)\}$.
- "combine" on a subset U of vertices returns a new vertice, which combines all vertices in U with interleaving operator. For example, let $U = \{a^*, b^+\}$, combine(U) is $a^*\&b^+$.
- "clique_removal" on an undirected graph G returns a maximum independent set (MIS). Finding an MIS of a graph G is a NP-hard problem. Hence we use the method clique_removal() [12] to find an approximate result.

The algorithm Soa2Soire is given in Algorithm 2. The main procedures are as follows.

(1) We first deal with all strongly connected looped components, replace each with a new vertex.
(2) After the SOA is a directed acyclic graph (DAG), focus on the set F of all vertices which can be reached from the source directly, but not via other vertices; make sure that there are no vertices which can be reached directly and via other vertices (if necessary, add an auxiliary node labeled ε).
(3) Recurse on the sets of vertices exclusively reachable from a vertex in F and contract these sets to vertices labeled with the result of the recursion.
(4) Combine vertices in F with "or", recurse again on what is exclusively reachable from this new vertex.
(5) Once only one item is left in F, split it off and recurse on the remainder.

Note that the algorithm introduces "?" by way of constructing "or ε". This can be cleaned up by postprocessing the resulting SOIRE.

The algorithm Merge is given in Algorithm 3. The main procedures are as follows.

(1) The first step (line 1): We first compute the constraint set *constraint_tr* using the function cs(S).
(2) The second step (line 4): We construct an undirected graph G using element in *constraint_tr* as edges.
(3) The third step (lines 5-8): We select a maximum independent set (MIS) of G, add it to list *all_mis* and

Algorithm 2: Soa2Soire

```
Input: a set of positive sample $S$; an SOA $\mathcal{A} = (V,E)$
Output: an SOIRE
if $|E| = 0$ then return $\varnothing$;
else if $|V| = 2$ then return $\varepsilon$;
else if $\mathcal{A}$ has a cycle then
    Let $U$ be a strongly connected component of $\mathcal{A}$;
    if $|U| = 1$ then
        Let $v$ be the only vertice of $U$;
        $\mathcal{A}$.contract($U$,plus($v$.label()));
    else $\mathcal{A}$.contract($U$,Merge(filter($U$, $S$)));
else if $\mathcal{A}$.succ($\mathcal{A}$.src) $\neq$ $\mathcal{A}$.first() then
    $\mathcal{A}$.addEpsilon();
else if $|\mathcal{A}$.first()$| = 1$ then
    Let $v$ be the only successor of $src$;
    $\delta \leftarrow v$.label();
    $\mathcal{A}$.contract($\{\mathcal{A}$.src,$v\}$,src);
    $\delta' \leftarrow$ Soa2Soire($S$,$\mathcal{A}$);
    return concatenate($\delta$,$\delta'$);
else if $\exists v \in \mathcal{A}$.first(), $\mathcal{A}$.exclusive($v$) $\neq \{v\}$ then
    Let $v$ be such that $\mathcal{A}$.exclusive($v$) $\neq \{v\}$;
    $U \leftarrow \mathcal{A}$.exclusive($v$);
    $\mathcal{A}$.contract($U$,Soa2Soire($S$,$\mathcal{A}$.extract($U$)));
else
    Let $u$,$v \in \mathcal{A}$.first() with $u \neq v$ s.t. $\mathcal{A}$.reach($u$) $\cap$
      $\mathcal{A}$.reach($v$) is $\subseteq$-maximal;
    $\mathcal{A}$.contract($\{u$,$v\}$,or($u$.label(),$v$.label()));
return Soa2Soire($S$,$\mathcal{A}$);
```

Algorithm 3: Merge

```
   Input: a set of positive sample $S$
   Output: an epression $\zeta$
1  $constraint\_tr \leftarrow$ cs($S$);
2  $U \leftarrow \varnothing$;
3  $G \leftarrow$ Graph($constraint\_tr$);
4  $all\_mis \leftarrow \varnothing$;
5  while $|G.nodes()| > 0$ do
6      $W \leftarrow$ clique_removal($G$) [12];
7      $G \leftarrow G \setminus W$;
8      $all\_mis.append(G)$
9  foreach $mis \in all\_mis$ do
10     $S' \leftarrow$ filter($mis$, $S$)
11     Construct SOA $\mathcal{A}$ for $S'$ using method 2T-INF [20];
12     $\delta \leftarrow$ Soa2Soire($S'$,$\mathcal{A}$)
13     $U$.append($\delta$)
14 return $\zeta \leftarrow$ combine($U$)
```

delete the MIS and their related edges from G. The process is repeated until there exists no nodes in G.

(4) The fourth step (lines 9-13): We get the sample set S' using the function filter(mis, S) for each MIS, and construct SOAs for sample sets by calling the algorithm 2T-INF [20]. Then convert SOAs into SOIREs using algorithm Soa2Soire.

(5) The last step (line 14): We call the function combine to generate an expression ζ with interleaving operator.

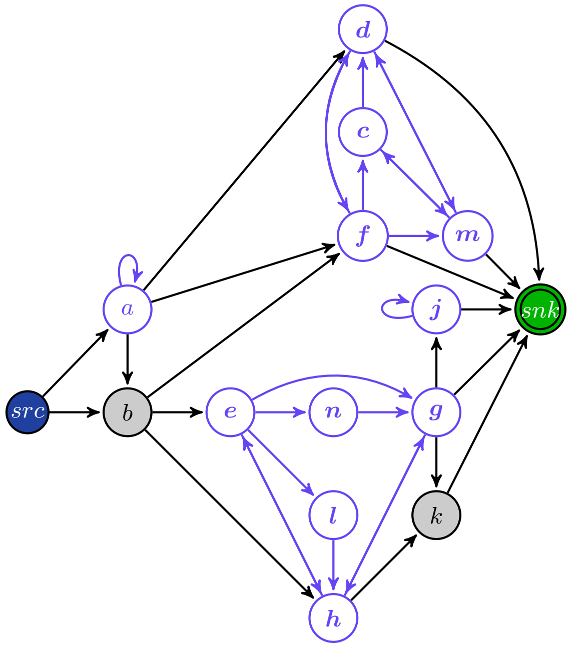


Figure 4: Four SCCs of SOA.

Following the example in section 3.1, there are four strongly connected components $U_1 = \{a\}$, $U_2 = \{j\}$, $U_3 = \{f, d, m, c\}$ and $U_4 = \{l, g, h, e, n\}$ shown in Figure 4. For strongly connected component (SCC) $U_1 = \{a\}$, because $|U_1| = 1$, we use $\mathcal{A}$.contract(U_1,plus(j)) to modify $\mathcal{A}$ such that vertice a is contracted to a new vertex a^+ and the self-loop is removed. Similarly, we use $\mathcal{A}$.contract(U_2,plus(j)) to modify $\mathcal{A}$ such that vertice j is contracted to a new vertex j^+ and the self-loop is removed (Figure 5). For SCC U_3, because $|U_3| > 1$, so we should call $\mathcal{A}$.contract(U_3,Merge(filter(U_3, S))). In this sub-process, we first compute the new sample set S_1=$\{fmcd, fcmd, df, fdm\}$ using function filter(U_3,S). Then we get cs(S_1) = $\{\langle f, d\rangle, \langle d, f\rangle, \langle m, d\rangle, \langle d, m\rangle, \langle m, c\rangle, \langle c, m\rangle\}$ in the algorithm Merge. Next, we constructing undirected graph G_1 based on cs(S_1) shown in Figure 6. We compute the set of all maximum independent sets (all_mis = $\{\{f, m\}, \{c, d\}\}$) for Figure 6. We construct two SOAs using filter($\{f, m\}$, S_1) and filter($\{c, d\}$, S_1), respectively. They are shown in Figure 7 and Figure 8. We convert two SOAs into $fm^?$ and $c^?d$, respectively. Then we get the new label $\zeta = fm^?\&c^?d$ using combine($fm^?$,$c^?d$). We use $\mathcal{A}$.contract(U_3,ζ) to modify $\mathcal{A}$ such that all vertices of U_3 are contracted to a single vertex and labeled ζ (edges are moved accordingly) shown in Figure 9. Similarly, we also call $\mathcal{A}$.contract(U,Merge(filter(U_4, S))). We first compute the new sample set $S_2 = \{egh, eng, eg,$

$elhg, ehg, heg\}$ using filter(U_4,S). Then we get cs(S_2) $= \{\langle g,h\rangle, \langle h,g\rangle, \langle h,e\rangle, \langle e,h\rangle\}$ in the algorithm Merge. Next, we constructing undirected graph G_2 based on cs(S_2) shown in Figure 10. We compute the set of all maximum independent sets $\{\{l,g,e,n\},\{h\}\}$ for Figure 10. We construct two SOAs using filter($\{l,g,e,n\}$, S_2) and filter($\{h\}$, S_2), respectively. They are shown in Figure 11 and Figure 12. We convert two SOAs into $e(n|l)^?g$ and $h^?$, respectively. Then we get the new label $\delta = e(n|l)^?g\&h^?$ using combine($e(n|l)^?g$,$h^?$). We use $\mathcal{A}$.contract(U_4,δ) to modify $\mathcal{A}$ such that all vertices of U_4 are contracted to a single vertex and labeled δ (edges are moved accordingly) shown in Figure 13. Continue to execute the remaining processes of the algorithm iSOIRE and we get the final inferred result r=$a^*b^?(fm^?\&c^?d|e(n|l)^?g\&h^?)(j^+|k)^?$.

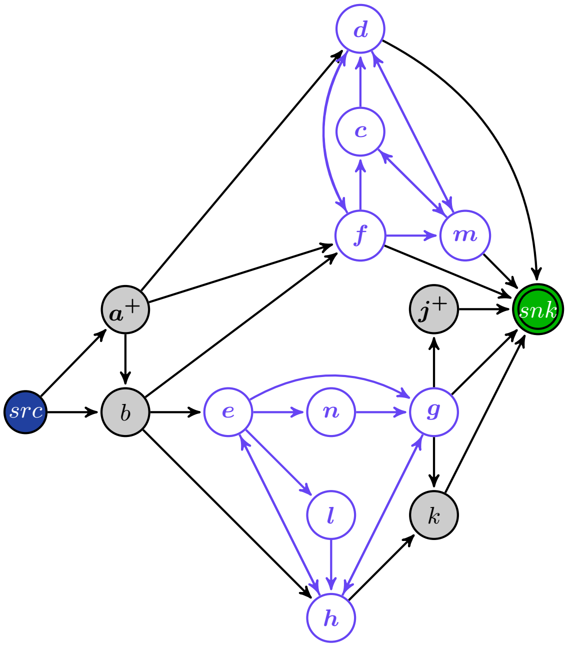


Figure 5: Dealing with SCC U_1 and U_2 of SOA $\mathcal{A}$.

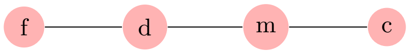


Figure 6: Constructing undirected graph G_1.

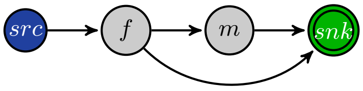


Figure 7: Constructing SOA $\mathcal{A}_1$ of filter($\{f,m\}$, S_1).

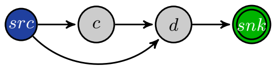


Figure 8: Constructing SOA $\mathcal{A}_2$ of filter($\{c,d\}$, S_1).

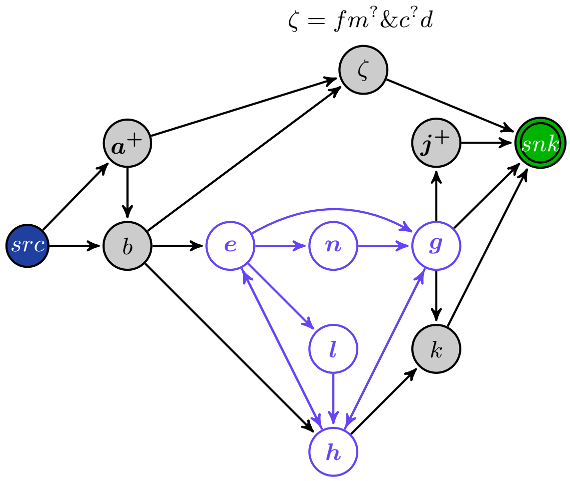


Figure 9: Dealing with SCC U_3 of SOA $\mathcal{A}$.

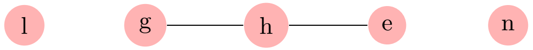


Figure 10: Constructing undirected graph G_2.

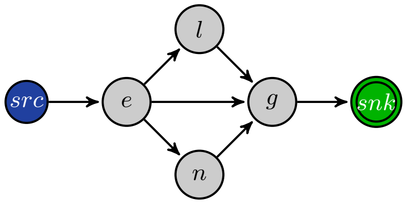


Figure 11: Constructing SOA $\mathcal{A}_3$.

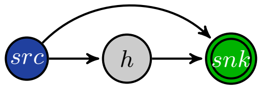


Figure 12: Constructing SOA $\mathcal{A}_4$.

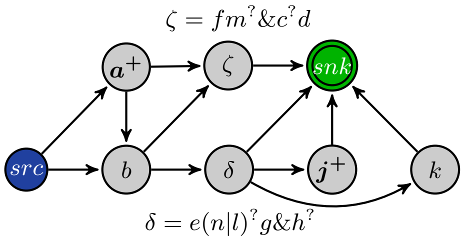


Figure 13: Dealing with SCC U_4 of SOA $\mathcal{A}$.

4 EXPERIMENTS

In this section, we conduct a series of experiments to analyze the practicability of SOIRE, and compare algorithm *i*SOIRE with not only the learning algorithms from ongoing researches but also the industrial-level tools used in real world. In terms of *preciseness* and *conciseness*, our work has achieved satisfying results compared with existing methods, reaching higher preciseness with less description length. Specifically, indicators *Language Size* ($|\boldsymbol{\mathcal{L}(r)}|$) [5] and *datacost* (**DC**) [5] are used to measure preciseness, while $\boldsymbol{\mathcal{L}en}$ [30] and Nesting Depth (**ND**) [31] for conciseness. Similar as the discussion of $|\mathcal{L}(r)|$ and $\mathcal{L}en$ above, we have that larger the value of DC (ND) is, more precise (concise) the regular expression will be. *Language Size* [5], denoted by $|\mathcal{L}(r)|$, is defined as:

$$|\mathcal{L}(r)| = \sum_{\ell=1}^{\ell_{max}} |L^{\ell}(r)|,$$

where $|L^{\ell}(r)|$ is the size of subset containing words with length ℓ in $L(r)$. Generally, $L(r)$ is an infinite language with infinitely large value of ℓ, it is of course impossible to take all words into account. Hence, we only consider the word length ℓ up to a maximum value: $\ell_{max} = 2m + 1$ where m is the length of r excluding ε, $\varnothing$ and regular expression operators. *Language Size* ($|\mathcal{L}(r)|$) can well measure the preciseness of a regular expression. Smaller the value of $|\mathcal{L}(r)|$ is, more precise the regular expression will be. *datacost* (**DC**) [5], is defined as:

$$datacost(r, S) = \sum_{\ell=1}^{\ell_{max}} \left(2 \times log_2 \ell + log_2 \left(\frac{|L^{\ell}(r)|}{|S^{\ell}|} \right) \right),$$

where $\ell_{max} = 2m+1$ and $|L^{\ell}(r)|$ as before, $|S^{\ell}|$ is the number of words in S that have length ℓ. Smaller the value of **DC** is, more precise the regular expression will be. $\mathcal{L}en$ [30] is defined as:

$$\mathcal{L}en = n \times \lceil \log_2(|\Sigma| + |\mathcal{M}|) \rceil,$$

where $|\Sigma|$ is the number of distinct symbols occurring in regular expression r, $\mathcal{M}$ is the set of metacharacters $\{|, \cdot, \&, ?, *, +, (,)\}$ and n is the length of r including symbols and metacharacters. An expression with a smaller value of $\mathcal{L}en$ is more concise. Nesting Depth (**ND**) [31] is defined as:

- $\text{ND}(r) = 0$, if $r = \varepsilon$, $\varnothing$ or a for $a \in \Sigma$.
- $\text{ND}(r) = \text{ND}(r_1) + 1$, if $r = r_1^*$, $r = r_1^?$ or $r = r_1^+$, where r_1 is a regular expression over Σ.
- $\text{ND}(r) = max\{\text{ND}(r_1),\text{ND}(r_2)\}$, if $r = r_1 | r_2$, $r = r_1 \cdot r_2$ or $r = r_1 \& r_2$, where r_1 and r_2 are regular expressions over Σ.

The learning algorithms compared in experiments are Soa2Sore [17] and Soa2Chare [17], GenEchare [16], $learner^{+}_{DME}$ [13], conMiner [40], GenICHARE [43] and GenESIRE [32]. The industrial tools which are capable of supporting inference of XML schemas used in this section include IntelliJ IDEA[1], Liquid Studio[2], Trang[3], and InstanceToSchema[4].

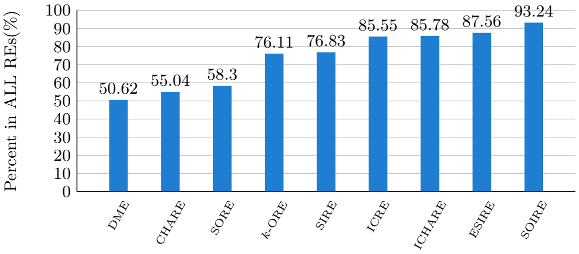


Figure 14: The proportion of subclasses on Relax NG. The dataset used for this statistical experiment is acquired from [28], with $509,267$ **regular expressions from** $4,526$ **Rleax NG schemas.**

For the massive comparative experiments, we conduct the experiments based on two kinds of datasets: small dataset (i.e., *mastersthesis*) and large dataset (i.e, *www*) of XML documents, which are both extracted from DBLP. DBLP is a data-centered database of information on major computer science journals and proceedings. We download the file of version *dblp-2015-03-02.xml.gz*[5]. *mastersthesis* and *www* are two elements chosen from DBLP with 5 (small) and 2,000,226 (large) samples, respectively.

All of our experiments are conducted on a machine with 16 cores Intel Xeon CPU E5620 @ 2.40GHz with 12M Cache, 24G RAM, OS: Windows 10.

4.1 Usage of SOIRE in Practice

Though interleaving is indispensable in data-centric applications, the lack of research on it is still a concern. In Figure 14, we visualized the coverage rates of regular expressions covered by different subclasses on Relax NG. We can see that the initial subclass, DME, only covers 50.62%. Then the proportions show an upward trend, reaching more than 85.55% (ICRE, ICHARE, ESIRE). Compared with their coverage, SOIRE covers 93.24%, which is 5.68% more than the second largest proportion. Therefore, the experimental result reveals the high practicality of SOIRE, and its strong support for interleaving.

4.2 Analysis of Inference Results

To better illustrate the performance of our work, we first compare the inferred results of our work with that of existing learning algorithms and industrial tools in real world. To save space, we use the short names of words and the list of abbreviations is shown in Table 1. The experimental results are shown in Table 2-5.

[1] https://www.jetbrains.com/idea/
[2] https://www.liquid-technologies.com/
[3] http://www.thaiopensource.com/relaxng/trang.html
[4] http://www.xmloperator.net/i2s/
[5] http://dblp.org/xml/release/dblp-2015-03-02.xml.gz

Table 1: The list of abbreviations for words in DBLP.

Word	Abbr.	Word	Abbr.	Word	Abbr.
author	a	editor	b	title	c
booktitle	d	pages	e	year	f
address	g	journal	h	volume	i
number	j	month	k	url	l
ee	m	cdrom	n	cite	o
publisher	p	note	q	crossref	r
isbn	s	series	t	school	u
chapter	v	publnr	w		

We can see from Table 3 that for dataset *mastersthesis*, the first six algorithms/tools (Liquid Studio, Soa2Sore, Soa2Chare, GenEchare, IntelliJ IDEA and Trang) reach high conciseness at enormous cost of $|\mathcal{L}(r)|$, from unaffordable 1.57×10^{10} to 1.64×10^{4}. Algorithms/tools InstanceToSchema, $learner^{+}_{DME}$ and conMiner have highest conciseness, with 52 for $\mathcal{L}en$, yet their preciseness is not the highest among these algorithms. Finally, the last three algorithms including *i*SOIRE reach the performance at the same level, with highest preciseness and the equal magnitude of conciseness. From the table we can draw a conclusion that though interleaving could improve the preciseness, the former one sacrifices the conciseness to some degree.

Table 2: Expressions of inference using different learning algorithms/inference tools on mastersthesis.

Method	Regular Expression
Liquid Studio	$(a\mid c\mid f\mid u\mid l\mid m)^{+}$
Soa2Sore	$acfu(l\mid m)^{*}$
Soa2Chare	$acfu(l\mid m)^{*}$
GenEchare	$acfu(l\mid m)^{*}$
IntelliJ IDEA	$acfu(l\mid m)^{*}$
Trang	$acfu(l\mid m)^{*}$
InstanceToSchema	$a\&c\&f\&l^{?}\&m^{?}\&u$
$learner^{+}_{DME}$	$a\&c\&f\&l^{?}\&m^{?}\&u$
conMiner	$acful^{?}\&m^{?}$
GenICHARE	$acfu(l^{?}\&m^{?})$
GenESIRE	$acfu(l^{?}\&m^{?})$
*i*SOIRE	$acfu(l^{?}\&m^{?})$

Table 3: Results of inference using different learning algorithms/inference tools on mastersthesis.

Method	$\lvert\mathcal{L}(r)\rvert$	DC	$\mathcal{L}en$	ND
Liquid Studio	1.57×10^{10}	122.880	56	1
Soa2Sore	1.64×10^{4}	67.657	56	1
Soa2Chare	1.64×10^{4}	67.657	56	1
GenEchare	1.64×10^{4}	67.657	56	1
IntelliJ IDEA	1.64×10^{4}	67.657	56	1
Trang	1.64×10^{4}	67.657	56	1
InstanceToSchema	984	102.446	52	1
$learner^{+}_{DME}$	984	102.446	52	1
conMiner	13	72.886	52	1
GenICHARE	**5**	**65.072**	**60**	**1**
GenESIRE	**5**	**65.072**	**60**	**1**
***i*SOIRE**	**5**	**65.072**	**60**	**1**

For the second dataset (Table 5), the advantage of our work is more outstanding. Without supporting the usage of interleaving, the previous eleven algorithms/tools have huge $|\mathcal{L}(r)|$ and DC, from 1.11×10^{21} to 4.39×10^{11} and from 15158.773 to 8479.873, respectively. Among them, Liquid Studio, Soa2Chare and IntelliJ IDEA have the shortest $\mathcal{L}en$, which are 120, while $learner^{+}_{DME}$ and ESIRE have the longest, which are 175. Soa2Sore has the deepest ND [31], with 3, followed by Liquid Studio, GenEchare, GenICHARE and GenESIRE, with 2 nestings. On the other hand, the algorithms/tools which support interleaving have smaller values on average. Especially for the indicator $|\mathcal{L}(r)|$, the magnitudes are much smaller than that of the first group of methods. It is noteworthy that our work reaches almost the same conciseness with much less values of $|\mathcal{L}(r)|(1.84 \times 10^{11})$ and DC(7599.996).

Table 4: Expressions of inference using different learning algorithms/inference tools on www.

Method	Regular Expression
Liquid Studio	$(a\mid c\mid l^{+}\mid q^{+}\mid o\mid b\mid f\mid m\mid d\mid r)^{+}$
Soa2Sore	$b^{*}(a^{*}(c(m^{?}\mid d))^{?}(l\mid q\mid f\mid o)^{*})^{+}\mid r$
Soa2Chare	$b^{*}r^{?}(m\mid o\mid f\mid a\mid l\mid q\mid c\mid d)^{*}$
GenEchare	$(b^{+}\mid r)^{?}(m\mid o^{+}\mid f\mid a^{+}\mid l^{+}\mid q^{+}\mid c\mid d)^{*}$
IntelliJ IDEA	$r^{?}b^{*}(a\mid d\mid o\mid m\mid q\mid c\mid l\mid f)^{*}$
Trang	$b^{*}(r\mid (a\mid d\mid o\mid m\mid q\mid c\mid l\mid f)^{+})$
InstanceToSchema	$m^{?}\&q^{*}\&b^{*}\&f^{?}\&a^{*}\&o^{*}\&c^{?}\&d^{?}\&r^{?}\&l^{+}$
$learner^{+}_{DME}$	$(q^{*}\mid f^{?}\mid r^{?})\&(o^{*}\mid d^{?}\mid m^{?})\&(a^{*}\mid b^{*})\&c^{*}\&l^{*}$
conMiner	$r^{?}b^{*}c^{*}o^{*}d^{?}m^{?}f^{?}\&a^{*}q^{*}\&l^{*}$
GenICHARE	$(b^{+}\mid r)^{?}(a^{*}q^{*}d^{?}m^{?}\&c^{+}o^{*}f^{?}\&l^{*})^{?}$
GenESIRE	$(b^{+}\mid r)^{?}(a^{*}(m^{?}\mid q^{*}\mid d)\&c(o^{*}\mid f)\&l^{*})^{?}$
*i*SOIRE	$b^{*}((a^{+}(q^{*}\mid d^{?})\mid m)\&c(o^{*}\mid f)\&l^{*})\mid r$

It is clear from the above analysis, our work outperforms other state-of-the-art learning algorithms and published

Table 5: Results of inference using different learning algorithms/inference tools on www.

Method	$\lvert\mathcal{L}(r)\rvert$	DC	$\mathcal{L}en$	ND
Liquid Studio	1.11×10^{21}	15158.773	120	2
Soa2Sore	1.30×10^{12}	7190.139	165	3
Soa2Chare	1.36×10^{19}	13696.752	120	1
GenEchare	1.34×10^{19}	13685.703	150	2
IntelliJ IDEA	1.36×10^{19}	13696.752	120	1
Trang	1.20×10^{19}	13606.698	125	1
InstanceToSchema	1.53×10^{18}	13406.824	145	1
$learner^{+}_{DME}$	1.43×10^{15}	11150.850	175	1
conMiner	4.11×10^{13}	10453.822	145	1
GenICHARE	1.41×10^{13}	9961.492	170	2
GenESIRE	4.39×10^{11}	8479.873	175	2
***i*SOIRE**	$\mathbf{1.84 \times 10^{11}}$	**7599.996**	**165**	**1**

tools, achieving the highest preciseness and the equal level of conciseness. Furthermore, through the comparison, the performance of our method indicates that the involvement of interleaving could contribute to both preciseness and conciseness.

5 CONCLUSION AND FUTURE WORK

Based on large-scale real data, we proposed a new subclass SOIRE of regular expressions with interleaving. SOIRE is more powerful than the existing subclasses and has unrestricted support for interleaving. Correspondingly, we design an inference algorithm *i*SOIRE which can learn SOIREs effectively based on single occurrence automaton (SOA) and maximum independent set (MIS). We conduct a series of experiments, comparing the performance of our algorithm with both ongoing learning algorithms in academia and industrial tools in real-world. The results reveal the practicability of SOIRE and the effectiveness of *i*SOIRE, showing the high preciseness and conciseness of our work.

We will study another subclass of regular expressions: k-occurrence regular expressions with interleaving (k-OIREs) in our future work. Its inference algorithm will also be considered.

6 ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61872339 and 61472405.

REFERENCES

[1] Serge Abiteboul, Pierre Bourhis, and Victor Vianu. 2015. Highly Expressive Query Languages for Unordered Data Trees. *Theory Comput. Syst.* 57, 4 (2015), 927–966.

[2] Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, and Carlo Sartiani. 2019. Parametric schema inference for massive JSON datasets. *The VLDB Journal* (Jan 2019).

[3] Denilson Barbosa, Laurent Mignet, and Pierangelo Veltri. 2005. Studying the XML Web: Gathering Statistics from an XML Sample. *World Wide Web* 8, 4 (2005), 413–438.

[4] Martin Berglund, Henrik Björklund, and Johanna Björklund. 2013. Shuffled languages - Representation and recognition. *Theor. Comput. Sci.* 489-490 (2013), 1–20.

[5] Geert Jan Bex, Wouter Gelade, Frank Neven, and Stijn Vansummeren. 2010. Learning Deterministic Regular Expressions for the Inference of Schemas from XML Data. *TWEB* 4, 4 (2010), 14:1–14:32.

[6] Geert Jan Bex, Frank Neven, and Jan Van den Bussche. 2004. DTDs versus XML Schema: A Practical Study. In *Proceedings of the Seventh International Workshop on the Web and Databases, WebDB 2004, June 17-18, 2004, Maison de la Chimie, Paris, France, Colocated with ACM SIGMOD/PODS 2004*. 79–84.

[7] Geert Jan Bex, Frank Neven, Thomas Schwentick, and Karl Tuyls. 2006. Inference of Concise DTDs from XML Data. In *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*. 115–126.

[8] Geert Jan Bex, Frank Neven, Thomas Schwentick, and Stijn Vansummeren. 2010. Inference of Concise Regular Expressions and DTDs. *ACM Transactions on Database Systems* 35, 2 (2010), 1–47.

[9] Geert Jan Bex, Frank Neven, and Stijn Vansummeren. 2007. Inferring XML Schema Definitions from XML Data. In *Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, Austria, September 23-27, 2007*. 998–1009.

[10] Mikolaj Boja'nczyk, Anca Muscholl, Thomas Schwentick, Luc Segoufin, and Claire David. 2006. Two-Variable Logic on Words with Data. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*. 7–16.

[11] Iovka Boneva, Radu Ciucanu, and Slawek Staworko. 2013. Simple Schemas for Unordered XML. In *Proceedings of the 16th International Workshop on the Web and Databases 2013, WebDB 2013, New York, NY, USA, June 23, 2013*. 13–18.

[12] R Boppana and M M Halldrsson. 1992. Approximating Maximum Independent Set by Excluding Subgraphs. *Bit Numerical Mathematics* 32, 2 (1992), 180–196.

[13] Radu Ciucanu and Slawek Staworko. 2013. Learning Schemas for Unordered XML. In *Proceedings of the 14th International Symposium on Database Programming Languages (DBPL 2013), August 30, 2013, Riva del Garda, Trento, Italy.*

[14] James Clark and MURATA Makoto. 2003. RELAX NG Tutorial. Retrieved February 28, 2018 from https://relaxng.org/tutorial-20030326.html

[15] Dario Colazzo, Giorgio Ghelli, and Carlo Sartiani. 2011. Schemas for safe and efficient XML processing. In *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany*. 1378–1379.

[16] Xiaoqiang Feng, Lixiao Zheng, and Haiming Chen. 2014. *Inference Algorithm for a Restricted Class of Regular Expressions*. Vol. 41. Computer Science. 178–183 pages.

[17] Dominik D. Freydenberger and Timo Kötzing. 2015. Fast Learning of Restricted Regular Expressions and DTDs. *Theory Comput. Syst.* 57, 4 (2015), 1114–1158.

[18] Enrico Gallinucci, Matteo Golfarelli, and Stefano Rizzi. 2018. Schema profiling of document-oriented databases. *Inf. Syst.* 75 (2018), 13–25.

[19] Shudi Gao, Black Mesa C. M. Sperberg-McQueen, and Henry S. Thompson. 2012. W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures. Retrieved February 28, 2018 from https://www.w3.org/TR/xmlschema11-1/

[20] P. Garcia and E. Vidal. 2002. Inference of k-Testable Languages in the Strict Sense and Application to Syntactic Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 9 (2002), 920–925.

[21] Vijay K. Garg and M. T. Ragunath. 1992. Concurrent Regular Expressions and Their Relationship to Petri Nets. *Theor. Comput. Sci.* 96, 2 (1992), 285–304.

[22] Minos N. Garofalakis, Aristides Gionis, Rajeev Rastogi, S. Seshadri, and Kyuseok Shim. 2003. XTRACT: Learning Document Type Descriptors from XML Document Collections. *Data Min. Knowl. Discov.* 7, 1 (2003), 23–56.

[23] Jay L. Gischer. 1981. Shuffle Languages, Petri Nets, and Context-Sensitive Grammars. *Commun. ACM* 24, 9 (1981), 597–605.

[24] E. Mark Gold. 1967. Language Identification in the Limit. *Information and Control* 10, 5 (1967), 447–474.

[25] Steven Grijzenhout and Maarten Marx. 2013. The quality of the XML Web. *J. Web Semant.* 19 (2013), 59–68.

[26] Johanna Högberg and Lisa Kaati. 2010. Weighted unranked tree automata as a framework for plan recognition. In *13th Conference on Information Fusion, FUSION 2010, Edinburgh, UK, July 26-29, 2010*. 1–8.

[27] Marco Kuhlmann and Giorgio Satta. 2009. Treebank Grammar Techniques for Non-Projective Dependency Parsing. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*. 478–486.

[28] Yeting Li, Xinyu Chu, Xiaoying Mou, Chunmei Dong, and Haiming Chen. 2018. Practical Study of Deterministic Regular Expressions from Large-scale XML and Schema Data. In *Proceedings of the 22nd International Database Engineering & Applications Symposium, IDEAS 2018, Villa San Giovanni, Italy, June 18-20, 2018*. 45–53.

[29] Yeting Li, Chunmei Dong, Xinyu Chu, and Haiming Chen. 2019. Learning DMEs from Positive and Negative Examples. In *Database Systems for Advanced Applications - DASFAA 2019 International Workshops: BDMS, BDQM, and GDMA, Chiang Mai, Thailand, April 22-25, 2019, Proceedings*. 434–438.

[30] Yeting Li, Xiaoying Mou, and Haiming Chen. 2018. Learning Concise Relax NG Schemas Supporting Interleaving from XML Documents. In *Advanced Data Mining and Applications - 14th International Conference, ADMA 2018, Nanjing, China, November 16-18, 2018, Proceedings*. 303–317.

[31] Yeting Li, Xiaolan Zhang, Feifei Peng, and Haiming Chen. 2016. Practical Study of Subclasses of Regular Expressions in DTD and XML Schema. In *Web Technologies and Applications - 18th Asia-Pacific Web Conference, APWeb 2016, Suzhou, China, September 23-25, 2016. Proceedings, Part II*. 368–382.

[32] Yeting Li, Xiaolan Zhang, Han Xu, Xiaoying Mou, and Haiming Chen. 2018. Learning Restricted Regular Expressions with Interleaving from XML Data. In *Conceptual Modeling - 37th International Conference, ER 2018, Xi'an, China, October 22-25, 2018, Proceedings*. 586–593.

[33] Zheng Li and Tingjian Ge. 2015. PIE: Approximate interleaving event matching over sequences. In *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*. 747–758.

[34] Wim Martens, Frank Neven, Matthias Niewerth, and Thomas Schwentick. 2017. BonXai: Combining the Simplicity of DTD with the Expressiveness of XML Schema. *ACM Trans. Database Syst.* 42, 3 (2017), 15:1–15:42.

[35] Wim Martens, Frank Neven, and Thomas Schwentick. 2013. Complexity of Decision Problems for XML Schemas and Chain Regular Expressions. *Siam Journal on Computing* 39, 4 (2013), 1486–1530.

[36] Wim Martens, Frank Neven, Thomas Schwentick, and Geert Jan Bex. 2006. Expressiveness and complexity of XML Schema. *ACM Trans. Database Syst.* 31, 3 (2006), 770–813.

[37] Laurent Mignet, Denilson Barbosa, and Pierangelo Veltri. 2003. The XML web: a first study. In *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*. 500–510.

[38] Jun-Ki Min, Jae-Yong Ahn, and Chin-Wan Chung. 2003. Efficient extraction of schemas for XML documents. *Inf. Process. Lett.* 85, 1 (2003), 7–12.

[39] Joakim Nivre. 2009. Non-Projective Dependency Parsing in Expected Linear Time. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*. 351–359.

[40] Feifei Peng and Haiming Chen. 2015. Discovering Restricted Regular Expressions with Interleaving. In *Web Technologies and Applications - 17th Asia-PacificWeb Conference, APWeb 2015, Guangzhou, China, September 18-20, 2015, Proceedings*. 104–115.

[41] Arnaud Sahuguet. 2000. Everything You Ever Wanted to Know About DTDs, But Were Afraid to Ask (Extended Abstract). In *The World Wide Web and Databases, Third International Workshop WebDB 2000, Dallas, Texas, USA, Maaay 18-19, 2000, Selected Papers*. 171–183.

[42] Lanjun Wang, Oktie Hassanzadeh, Shuo Zhang, Juwei Shi, Limei Jiao, Jia Zou, and Chen Wang. 2015. Schema Management for Document Stores. *PVLDB* 8, 9 (2015), 922–933.

[43] Xiaolan Zhang, Yeting Li, Fanlin Cui, Chunmei Dong, and Haiming Chen. 2018. Inference of a Concise Regular Expression Considering Interleaving from XML Documents. In *Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II*. 389–401.

VISEN: A Video Interactive Retrieval Engine Based on Semantic Network in large video collections

Mohamed Hamroun
LABRI University of Bordeaux
MIRACL University of Sfax
France, Tunisia
mohamed.hamroun@u-bordeaux.fr

Sonia Lajmi
MIRACL University of Sfax
Al Baha University
Tunisia, Saudi Arabia
slajmi@bu.edu.sa

Henri Nicolas
LABRI University of Bordeaux
France
henri.nicolas@u-bordeaux.fr

Ikram Amous
MIRACL University of Sfax
Tunisia
ikram.amous@enetcom.usf.tn

ABSTRACT

Following technological advances carried out recently, there has been an explosion in the quantity of videos available and their accessibility. This is largely justified by the fall of the prices of acquisition and the increase of the capacity of the memory supports, which made the storage of the large document video in computer system possible. To allow an effective exploitation of the collections, it is necessary to install tools facilitating the access to the documents and handle them. In this context, we propose a multimedia retrieval approach that puts the user at the center of the retrieval process starting from a text query. The new aspects of our proposal is as follows: (i) concerning the indexation part, we propose a new approach allowing a multi-level and semantic classification of videos, (ii) regarding the retrieval part, the inclusion of query expansion mechanism helps the user to formulate the query and the relevance feedback mechanism which helps improve the results considering the user's feedback. Our contribution at the experimental level consists in the implementation of prototype VISEN. In fact the technique proposed have been integrated in system seeks by the contents to evaluate the contribution in terms of effectiveness and precision. After carrying out a set of tests on 2700 videos and 62838 images, the experimental results showed that the proposed algorithm performs well.

CCS CONCEPTS

• Multimedia Indexing • Multimedia Retrieval

KEYWORDS

Textual Query, Classification, Semantic Indexing, Relevance Feedback, Query Expansion, Concept, Context.

ACM Reference format:

Mohamed Hamroun, Sonia Lajmi, Henri Nicolas and Ikram Amous. 2019. VISEN: A Video Interactive Retrieval Engine Based on Semantic Network in large video collections. In IDEAS 2019: 23nd International Database Engineering & Applications Symposium, June 10–12, 2019, Athens, Greece. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331076.3331094

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331094

1. Introduction

Information retrieval consists of a set of operations to respond to the user's information needs via a GUI. First, the user must build a query. This obvious operation for the text is much more difficult for the image and the video queries. Indeed, the query can include different data: such as an image, a video, a sound, a drawing or an animation. The definition of queries is considered among the raised problems when searching large databases. The fact that the expressed query does not always translate the need for information in the head of the user by a textual request, although this form of request remains the most preferred by users than other forms (images, video, sketches, etc.)[1]. In what follows, we present several forms of interrogation and retrieval that are found in the different prototypes in the literature. In general, there are two great retrieval approaches: the textual queries and the conceptual queries.

In the textual form, the queries remain limited and are generally associated with a specific category of visual documents, such as TV news. The most promising way in this context regarding the videos consists of transcribing the sound track to determine its subject [2][3], rather than exploiting the visual content.

The conceptual queries are the subject of many researches works. For example, the INFORMEDIA approach uses a limited set of high-level concepts to filter the textual query results [4]. This system also creates groups of key-frames [5] and uses the results of the speech recognition to trace the collections of key-frames at the re-associated geographic place on a map and combine this with the other visualizations to give the user an arrangement of the query result context. The method suggested in [6] is based on a process of semantic indexing. This system uses a big semantic lexicon in categories and threads to support the interaction. It also defines a space of visual similarity, a space of semantic similarity, a semantic thread space and browsers to exploit these spaces. It is worth mentioning that the VERGE approach [7] supports the following functions: (i) a high level of visual conceptual retrieval and (ii) a visual retrieval. This tool combines indexing, analysis and recovery techniques of diverse modalities (textual, visual and conceptual).

In recent years, video retrieval based on the semantic concept has attracted the attention of many researchers [8] [9]. We mean by the detection of semantic concepts, detected the presence or the absence of high-level concepts such as bus, forest or sky in the videos. In [10], the authors combine the concept matching, the query, the corpus and the matching of the content. In [11, 12], the authors propose a video retrieval system by viewing and navigating in concepts. The proposed navigation module is based on a semantic classification. In [13, 14], the authors propose a method for the indexation and retrieval of video sequences in a large video database, based on a weighting technique that calculates the degree of membership of a concept in the audio-visual field. In [15, 35], the authors suggest a graphical approach aimed at providing the users with the possibility to dynamically display and explore a query result space built upon a document repository including interconnected media objects. [16, 35] introduces an interactive video browsing system based on content and developed for the Video Brower Showdown 2016. The aim of this system is to help the user find specific video clips among a large collection of videos within limited time frames.

Although many research efforts have been devoted to the detection of concepts, this task remains very difficult [17]. Most of the time, the problem is considered as a classification problem in which a binary classifier generally learns to predict the presence of a certain concept in a video sequence or key-frame based on the extracted feature descriptors.
Traditional content-based video search methods could not respond alone to the user's needs and have shown their limitations despite the great efforts for improvement [26]. In this context, Etter [19] improved his video search system by focusing on expanding the queries. To reach this goal, he used external data, such as Wikipedia content (i.e, titles and images). On the other hand, Elleuch et al. [20] implemented three automatic search subsystems which consist in extracting texts from videos, detecting visual and audio features. In [21], the authors developed a content-based video retrieval system to extract the color, texture and shape. First, the texture feature was extracted using the multi-fractal Brownian motion (mbm), then, the color feature was obtained using semantic color model and finally, the shape feature was extracted using the level set method. Then, all these features were stored in the indexing format. On the other hand, the IMOTION system [22] is a multimodal content-based video search and browsing application offering an arch set of query modes on the basis of a broad range of different features that can scale with the size of the collection due to its underlying flexible polystore called ADAM pro and its retrieval engine Cineast, for multi-feature fusion. In [25], the authors presented the semantic video indexing system of REGIMVid group to semantically access the multimedia archives, called SVI_REGIMVid, which is a generic approach for video indexing.
As indicated above, many research works have been carried out in the large audiovisual retrieval domain and suggested some tools based on diverse retrieval forms. These techniques have been formulated for the indexation of a video by its low-level content. Most of these studies put forward a very limited and specific study framework by proposing an approach based on a single modality. . This is justified by the fact that these media are related to fairly specific fields and sometimes very far. However, this attempt has not yet led to satisfactory results. On the other hand, semantic content indexing based on a combination of characteristics from different domains is innovative. Indeed, since the video components complement one another, the video will be semantically efficient

Moreover, these techniques have other limitations which are often related to the interaction with the user. Therefore, enriching the retrieval techniques with the user's past behavior potentially helps provide more pertinent results. In the literature, few research works have been interested in this aspect. For instance, the approach suggested in [26], was the first to put the user at the center of the retrieval process. In fact, the execution of a semi-automatic approach, which put the user at the center of the retrieval method, is a real improvement perspective of the existing methods.

Actually, the method suggested in our paper lies in this context as it proposes a large video retrieval approach starting from a textual query. Therefore, it can be said that our study has brought some contributions, such as (i) promoting an indexation approach based on DCM (data clustering method), (ii) providing a relevance feedback mechanism which consists in putting the user at the center of the retrieval process (iii) offering a query expansion mechanism which helps reformulate the textual query.

The remaining part of this paper is organized as follows: The system framework is described in sections 2, 3 and 4, the experimental setup and the results are presented in section 5, and finally, in section 6, we conclude with a summary and perspective of our works research..

2. General Description Of Proposed System

Fig.1 presents the global architecture of the suggested system VISEN (VIdeo System Engine). Like any information retrieval system, our VISEN system is decomposed according to the following functional phases:

Indexing Phase: Indexing is intended to extract and represent the meaning of a document so that it can be retrieved by the user. To reach this goal, we have suggested a classification approach called "DCM, Data Clustering Method" detailed in section 3.

Retrieval Phase: This phase consists in exploiting the result of the indexation phase besides, it includes some sub-phases, such as (i) a Query expansion phase which guides the user to rephrase his query in terms of concepts. The innovation of our method is the way to build the concepts from the entered keywords. To do this, we use the ontology and a descriptor vector that will be explained later. (ii) a Relevance Feedback which consists in the interaction with the user when the results are displayed. Indeed, several research works are proposed for the relevance feedback in the context of image and text information retrieval but rarely for the video filing. Inspired by the relevance feedback method initially proposed by [27], we propose a new method that can be applied to video documents based on concepts.

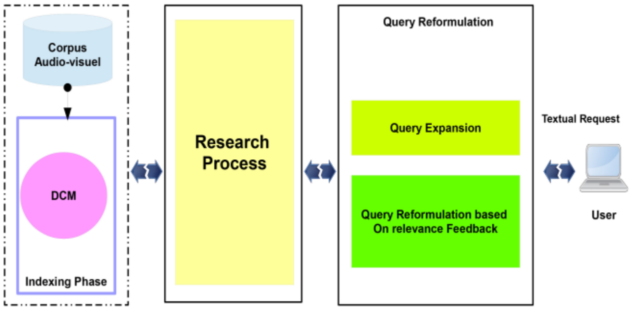


Figure 1: Conceptual Architecture of VISEN System

3. DCM An Efficient Data Clustering Method

For a relevant indexing, we propose a level consideration in the indexes. The organization is made according to three levels (contextual, conceptual, raw data) to facilitate its use during the retrieval phase. The idea is to associate the data and the common features that have a semantic relationship. In other words, videos with a common concept will be assigned to the same group.

Moving to a higher level of abstraction helps us organize the data and make it easier to be accessed. It is possible to semantically group similar concepts under the same context. Indeed, the navigation process can be done at the three following abstract levels:

Level 0: Contains all the audio-visual documents related to the different subjects;

Level 1: Contains concepts, like an Actor, a Boy, a Girl, a Face...;

Level 2: Contains the contexts representing the most relevant concepts in the corpus, like a person, an animal, a vehicle;

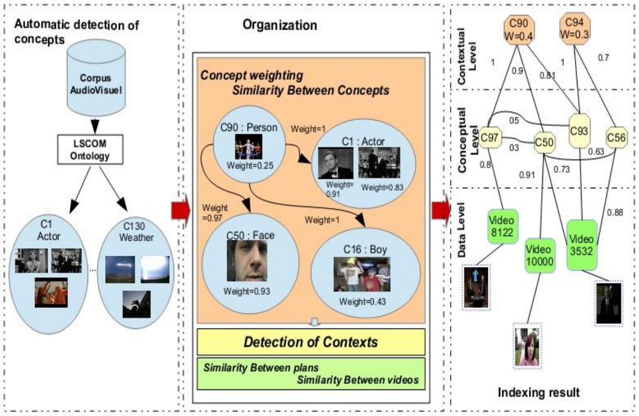


Figure 2: DCM Process

3.1 Automatic Detection Of Concepts

For the concept detection phase, we used the results of our semantic indexing system "SVI_LAMIRA"[1], which is based on our low-level descriptor “PMC” [28], on the one hand, and on our descriptor “PMGA” [28], on the other hand.

The result of this step enables us to move from Level 0 (Data) to Level 1 (Conceptual) as shown in figure 3.

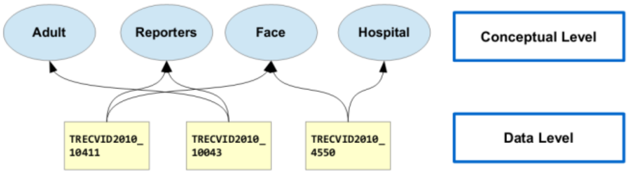


Figure 3: Automatic detection of concepts result

3.2 Weighting Of Concepts

In the previous section, the concepts have been organized and linked to video shots that they index. However, this concept-concept association is not weighted. In this step, to add a weighting to these concepts, on the one hand, and detect a context level to better structure and guide the navigation of the user, on the other hand.

Concepts weighting: In order to define the weights of concepts, a weight is associated for each video to each concept to value the importance in the video of each concept. Each video is therefore indexed by a set of concepts. For concept weighting, we adopt the TF-IDF (term frequency–inverse document frequency) measure. This measure is the most used IR work because it puts into perspective the importance of a concept in a document. For this reason, we propose a new measure that combines the local and global weights where the former depends only on a given video and the latter depends on the whole corpus. It should be noted that the weight of a concept C for a video V is given in the following equation:

$$W_{C,V} = TF\left(C_i, V_j\right).IDF\left(C_i, V_j\right) = \frac{Nbs\left(C_i, V_j\right)}{n} \frac{Nbcs\left(C_i\right)}{N.Nbc(V_j)} \tag{1}$$

$Nbs\left(C_i, V_j\right)$ is the number of shots containing Ci in V_j. n is the total number of shots in A. A is the number of identified concepts in V_j. $Nbc\left(V_j\right)$ is the number of videos containing Ci. $Nbcs\left(C_i\right)$ is the number of concepts similar to Ci.

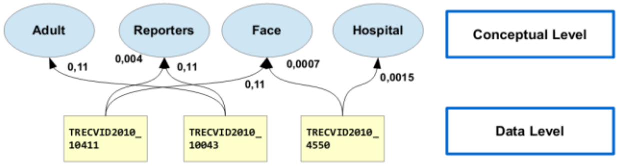


Figure 4: Weighting of concepts

3.3 Detection Of Contexts

Moving to another level of abstraction helps us to organize the data and speed up the access. We have grouped the more semantically similar concepts under the same context by proposing a method to extract the context from the concepts. Our context notion is inspired by the studies presented in the [28] which are based on the construction of a knowledge coding technology called topic-map. In fact, we adopted this technique in the audio-visual context by proposing the semantic entities called "context".

We define a context as a concept of which:

Is the most common appearance point of view in the audio-visual collection. The technique explained in section 3.2.1 is adopted to determine the frequency of this appearance. Therefore, the following equation is used:

$$E_1(i) = \sum_{1}^{N} P_{C_i, V_k} \tag{2}$$

$$i = \{concept\,1 ....Concept\,130\}$$

where E1 is the sum of concept weight Ci in all the Vk videos of the collection. N is the total number of concepts.

The total similarity with all the other concepts is the highest, therefore, the following equation is used:

$$E_2(i) = \sum_{1}^{N} Sim\left(C_i, C_j\right) \tag{3}$$

[1] Project of LaBRI Laboratory of the University of Bordeaux-France in collaboration with MIRACL laboratory of the University of Sfax-Tunisia.

Then, E2 is the sum of similarity values of Ci with all the concepts Cj of the ontology and N is the total number of concepts. The semantic relationship between the concepts will be detailed in section 3.2.3.

The combination of two weights gives us the following equation (2) * (3)

$$E(i) = \operatorname{Arg}\max(E_1(i).E_2(i)) = \operatorname{Arg}\max\left(\sum_{k}^{n} \operatorname{Sim}(C_i, C_j).\frac{\sum_{t}^{N} P_{C_i, V_k}}{N}\right) \quad (4)$$

The first term represents the relationship between all the concepts. The second term represents the frequency of the concepts in the corpus where E is the set of the selected contexts, n is the total number of concepts and N is the number of the videos containing Ci

This enables us to sort out a semantic summary (set of contexts) by indicating the importance of each concept in the collection. In addition, it provides a relevant starting point to help the user to begin the navigation process.
Figure 5 shows a global view of context detection process.

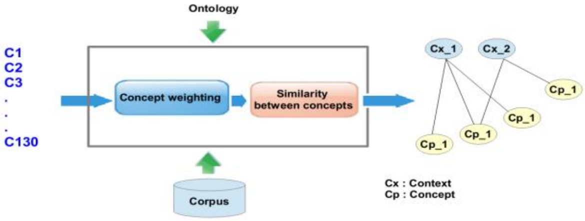


Figure 5: Context detection process

3.4 Inter-Concept Similarity

The semantic similarity distance used here is inspired by the distance of Rada which is based on the distance between two concepts in the ontology [29]. It is used to define a new measure of similarity between the concepts. Moreover, this new measure considers the amount of information shared between the two concepts in the corpus (the more the amount of shared information is important the more the concepts are similar).The definition of this measure is intended to serve as a transition between the contextual and conceptual levels. The similarity measure is therefore defined as:

$$Sim(Ci, Cj) = \frac{1}{dist(Ci, Cj)} \frac{\operatorname{Card}[\{Ci\} \cap \{Cj\}]}{\operatorname{Card}[\{Ci\} \cup \{Cj\}]} \quad (5)$$

The first term represents the probability to have both concepts when either of them is present. The second term shows that the similarity is reduced when the two concepts are far from each other

$\operatorname{Sim}(C_i, C_j)$ is the similarity between concepts Ci and Cj.

$\{C_i\} \cap \{C_j\}$ is the set of video shots in the whole corpus indexed with Ci and Cj.

$\{C_i\} \cup \{C_j\}$ is the set of video shots in the whole corpus indexed by Ci or Cj.

$\operatorname{dist}(C_i, C_j)$ is the distance between Ci and Cj which is equal to the number of links separating the two concepts in the ontology.

3.5 Similarity between videos we propose to classify the videos according to their semantic similarity: the more the videos have common concepts with weak differences between weight values, the more their content is semantically close. This is based on the Euclidean distance defined in the following equation.

$$\mathrm{D(Vi, Vj)} = \frac{1}{N}\sqrt{\sum_{\mathrm{K}=1}^{\mathrm{N}} \quad (W_{k,i}.W_{k,j})^2} \quad (6)$$

N: number of common concepts between V_i and V_j.
$W_{k,i}$: weight of the concept C_k in the video $V_{i.}$
$W_{K,j}$: weight of the concept C_k in the video $V_{j.}$

3.6 Similarity inter-plans we will adopt an approach that combines the measurement between a plans based on the arcs [13, 30] and the measurement of the informational content [31, 32]. For the informational content measurement, we have used the similarity of the cosine, while the measurement of the distance between the concepts is inspired by [13]. Through this combination, we can take advantages of both approaches at once. The equation is, then, defined as follows

$$\mathrm{Sim}(\mathrm{sh}_1, \mathrm{sh}_2) = \mathrm{Sim}(\mathrm{c_n}, \mathrm{c_m}) = \sum_{j=1}^{n} \frac{Pc_j(sh_1) * Pc_j(sh_2)}{\sqrt{\sum_{k=1}^{n}\left(Pc_k(sh_1)\right)^2} * \sqrt{\sum_{l=1}^{n}\left(Pc_l(sh_2)\right)^2}} + \frac{1}{1+\sum_{i=1}^{n}\sum_{j=1}^{m} dist(c_i(sh_1), c_j(sh_2)} \quad (7)$$

where $\mathrm{c_i(I_1)}$: The i concept in plan 1.
$\mathrm{Pc_j(I_1)}$: The concept j weight in plan 1
$\mathrm{dist(c_i(I_1), c_j(I_2)}$: The distance between concept Ci of plan 1 and concept Cj of plan I2 is the number of arcs separating both concepts.

In the end, as shown in figure 6, we proceed to generate a hierarchical structure made up of three levels: the first level represents the contexts; the second represents the concepts while the third represents the data.

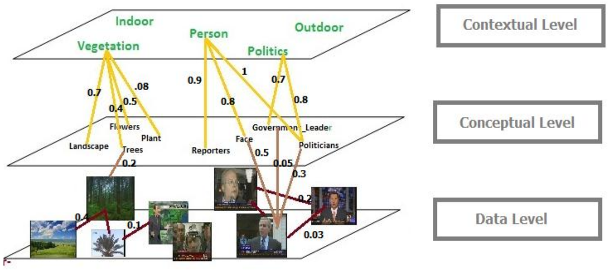


Figure 6: organization result

4. Search System Framework

Fig.7 shows the overall architecture of the proposed video search system. It breaks down according to the following functional phases:

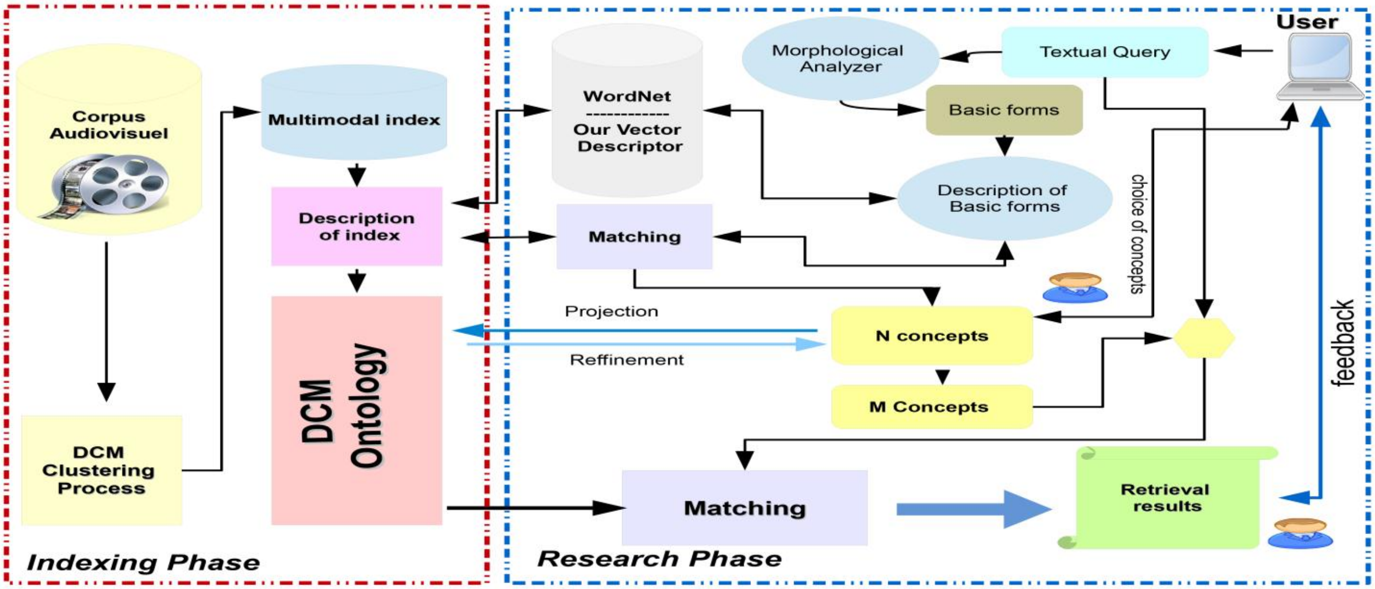


Figure 7: Conceptual Architecture of Search System “VISEN”

1. *Indexing phase (section 3)*

2. *Indexation of the query*: equiprobable weighting between terms.

3. *Query expansion*: Generally, the most intuitive way for a user is to express a text query with keywords that define the content of the videos is looking for. In this context, being based only on keywords for retrieval in Arabic, French and English is considered insufficient because the keywords used in the query can be compared to the documents in the database, differences on several levels, for example:

- Morphological variations
- Lexical variations (different words are used for the same meaning)
- Semantic variations

Subsequently, the system analyzes the text query and translates it into concepts. The problem at this level is how to transform a textual query into concepts. To solve this problem we propose to use the query expansion technique based on a domain ontology to help the user formulate his query. The use of ontology for the user’s query enrichment (expansion) can be a solution (among others) to resolve the problem of semantic variations. Indeed, ontology offers resources shaped as semantic relations, which results in the improvement of the research results. Furthermore, the use of a morphological analyzer can be sufficient for the resolution of the morphological and lexical variations.

It should be noted, however, that the pertinence of the results obtained by an IR system does not depend only on the matching process (between query/documents), but also on the pertinence of the query, hence, it is necessary to reformulate the query using two approaches in an IR system, one is direct and the other is indirect.

As part of the user's assistance by improving his query by using:

A morphological and lexical analyzer (in our case a descriptor vector) and a semantic resource (in our case Wordnet and our own DCM ontology) for direct reformulation..

4. *Description of DCM ontology concepts*: The same enrichment method was used (Vector descriptor and WordNet)

5. *Matching* by making a comparison between the query term description (after enrichment) and the terms that are assigned to describe each concept among the concepts of our ontology and return the concepts that have similar descriptors by using the Jaccard distance:

$$Jaccard(D_{rq}, D_c) = \frac{|D_{rq} \cap D_c|}{|D_{rq} \cup D_c|} \qquad (8)$$

where

D_{rq} are the terms that are assigned to describe those of the query.

Dc are the terms that are assigned to describe each concept of the ontology.

6. *Concept refining*: We make a projection on the DCM ontology to do a refining by using equation 8 of section 3.4, meaning that starting from the identified concepts in the first step, we work to identify other concepts that have semantic relations with the latter and that can seem important for the user.

7. *The user’s choice*: The user intervenes to manually choose the concepts matching his needs.

8. *Concept comparison*: Matching these concepts with those of the whole collection is carried out following the way of a vector model:

$$Simi(req, V_i) = \cos(\overrightarrow{req}, \overrightarrow{V_i}) = \sum_{j=1}^{n} \frac{Pc_j(V_i) * Pc_j(req)}{\sqrt{\sum_{k=1}^{n}(Pc_k(V_i))^2} * \sqrt{\sum_{k=1}^{n}(Pc_l(req))^2}} \qquad (9)$$

where :

v_i : Video i.

req : Request.

$Pc_j(v_i)$: The concept j weight in video i.

$Pc_j(req)$: The concept j weight in request.

9. *Retrieval results.*

10. *Relevance Feedback Method***:** The relevance feedback method presented in this section is based on Rocchio’s model [20] to rephrase the query. This model is summed up as follows: Either a set of information extraction operations initiated by an initial query Q_0 which is, then, modified according to the system product outlets. Q_1 is the obtained modified query which is the closest to the optimal user’s query. The efficiency of this process will depend on the query quality of the initial query and the level of convergence of the successive iterations towards an optimal query. The application of the Rocchio's formula on our context is as follows:

$$Q_1 = Q_0 + \alpha \sum_{i=1}^{n_1} \frac{P_i}{n_1} - \beta \sum_{j=1}^{n_2} \frac{NP_i}{n_2} \qquad (10)$$

where

Q1is the vector of the new query

Q0 is the vector of the initial query

P_i: is the vector of concepts matching the pertinent videos that are restituted and assessed

NP_j is the vector of concepts matching the non-pertinent videos that are restituted and assessed.

n1 is the number of restituted and assessed pertinent concept

n2 is the number of restituted and assessed non-pertinent concepts

α is a positive parameter of weighting the concepts of videos judged to be pertinent

β is a positive parameter of weighting the concepts of videos judged to be non-pertinent

Illustration of relevance feedback process

5. User interface

To clearly illustrate the features offered by our VISEN system, we present the following scenario:

Let us suppose that the user types "Person" as query. VISEN will search the concepts the most pertinent to this query. In order to refine the query, a matching between the description of the ontology concepts and the query term is carried out using the equation 8 (initial concept selection) and equation 5 (to refine) presented respectively in sections 3 and 4. For instance, the returned result is the following concepts: Actor, Adult, etc.

Let us also suppose that the user selects the concept 'Actor' among the concepts presented by VISEN. VISEN applies, then, equation 9 presented in section 4, in order to present the video search result. Fig.8 presents the search results ordered by level of pertinence.

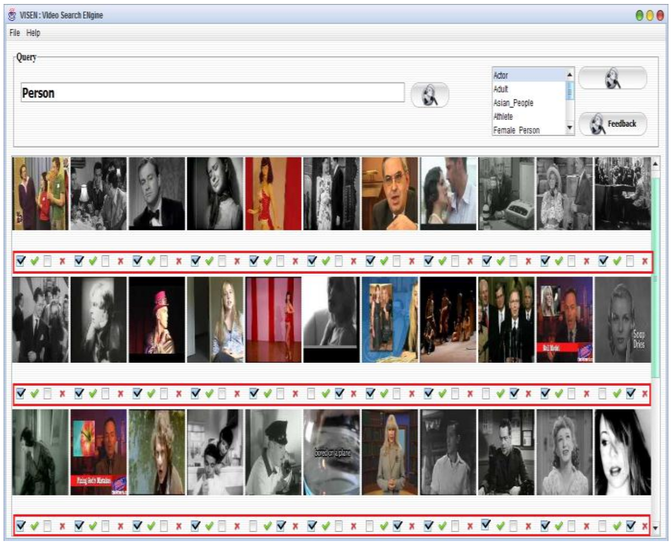


Figure 8: Result of the textual query

A relevance Feedback mechanism is allowed by the VISEN retrieval interface. In fact, in case of dissatisfaction with the result, the user will be able to choose those that are pertinent or non-pertinent for a given query. The system, then, recalculates the weight of the concepts by taking into consideration the user's indications and applying equation 10 presented in section 4.

By a simple click on the plan itself, the user can access the video relating to this plan. Similar images of the latter are displayed by using equation 6 and 7 of section 3.

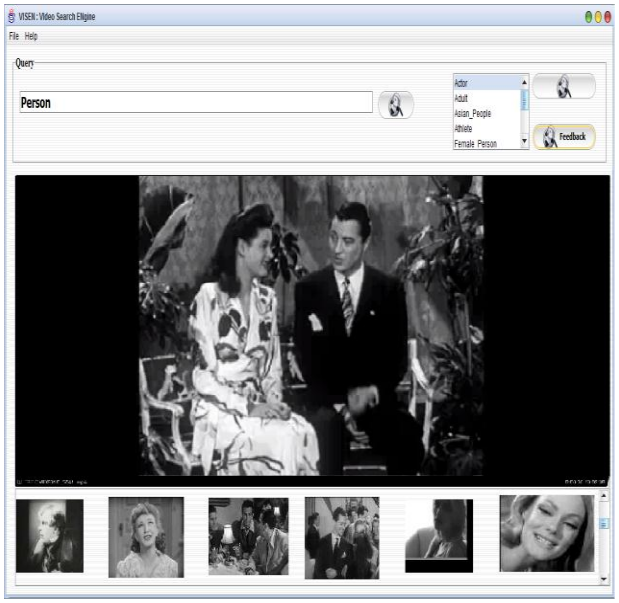


Figure 9: Audio-visual document player

6. Experimentation

6.1 Data Sets

At TRECVID 2015 Semantic Indexing task, there are two data sets provided by the National Institute of Standards and Technology (NIST): a test and a development data set. The development data set IACC.2.tv15 contains 3200 Internet Archive videos (50GB, 200 h) while the test data set IACC.2. contains approximately 8000 Internet Archive videos (50GB, 200 h). IACC.2. is annotated with 130 semantic concepts

The experimentation involves three main steps, which are: an experimentation of our VISEN prototype, an experimentation of the indexing (concept weighting pertinence) and an experimentation of the retrieval phase.

6.1 DCM Method

To better explain the Automatic detection of concepts results, we have presented them following the histogram presented in Fig.10, which represents the weighting level of the most pertinent concepts. These weighting levels are variable according to the systems. We have compared our proposed method to other works proposed. It is clear that the global concepts are the most pertinent ones (context) as they help cover the whole corpus.

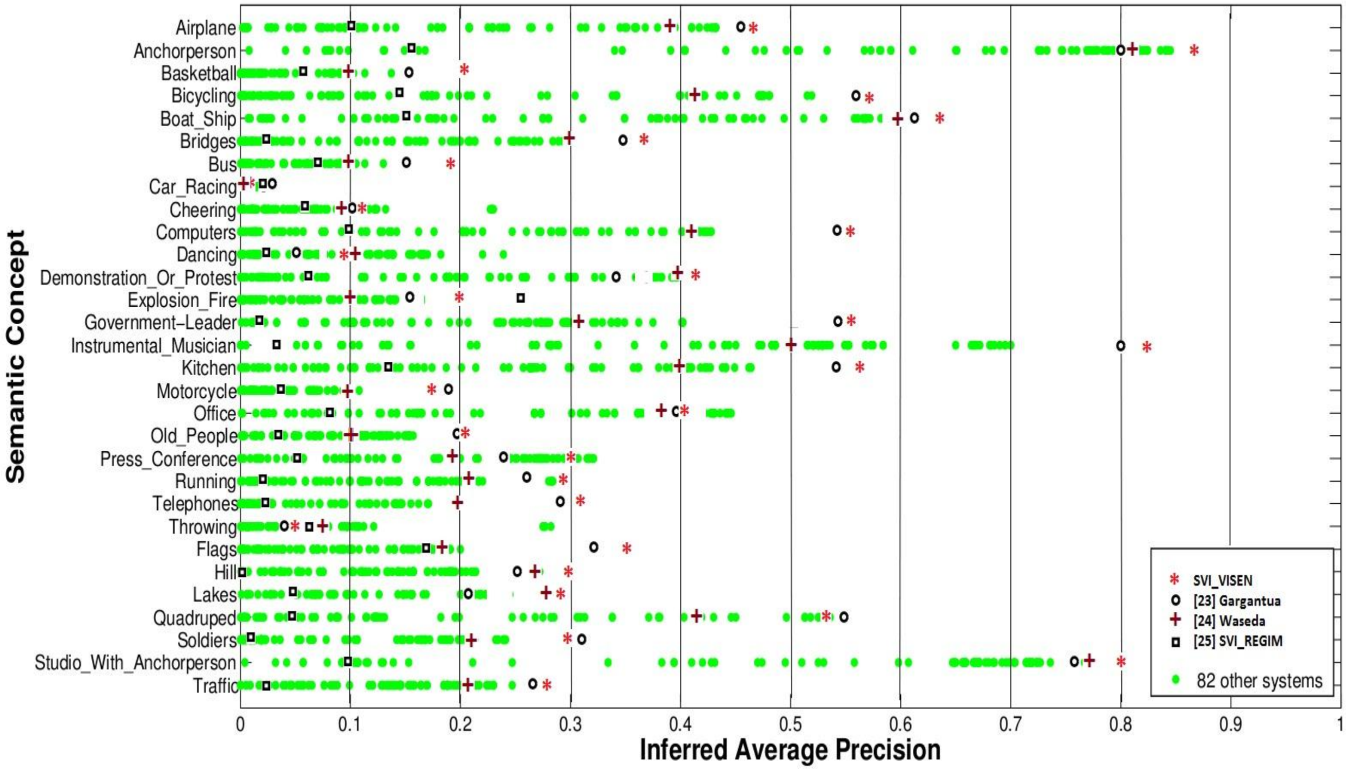


Figure 10: Comparison of the average precision of the [23],[24],[25] tools and proposed SVI_VISEN in the TRECVID 2015

6.2 Retrieval Phase

6.2.1 Comparison with other systems

In the 2nd step of the assessment, we will compare our system with the most pertinent semantic retrieval systems. The following figure (Fig.11) shows some query outcomes.

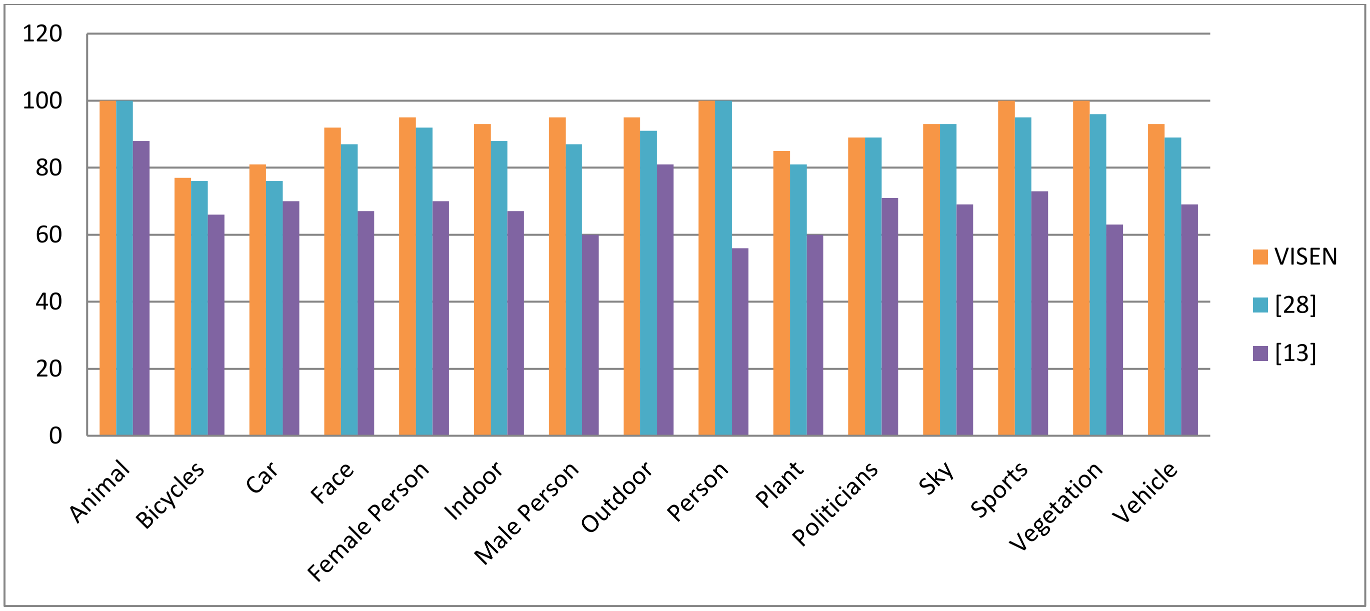


Figure 11: Precision value

Based on the histogram (Fig.11), we note that accuracy values corresponding to the sports and vegetation concepts are equal to 1. If compared to accuracy values corresponding to the works proposed by [13] and the one proposed by [28], we can see a significant improvement.

We observe that all the accuracy values exceed 0.85, which means that the improvement encompasses all the concepts. It is broadly clear that the suggested Interactive Search technique improves the system performance.

Fig.12 presents the comparison of interactive video search results for 24 topics performed by 36 users of the present-day video retrieval systems. The VISEN results are indicated with special markers.

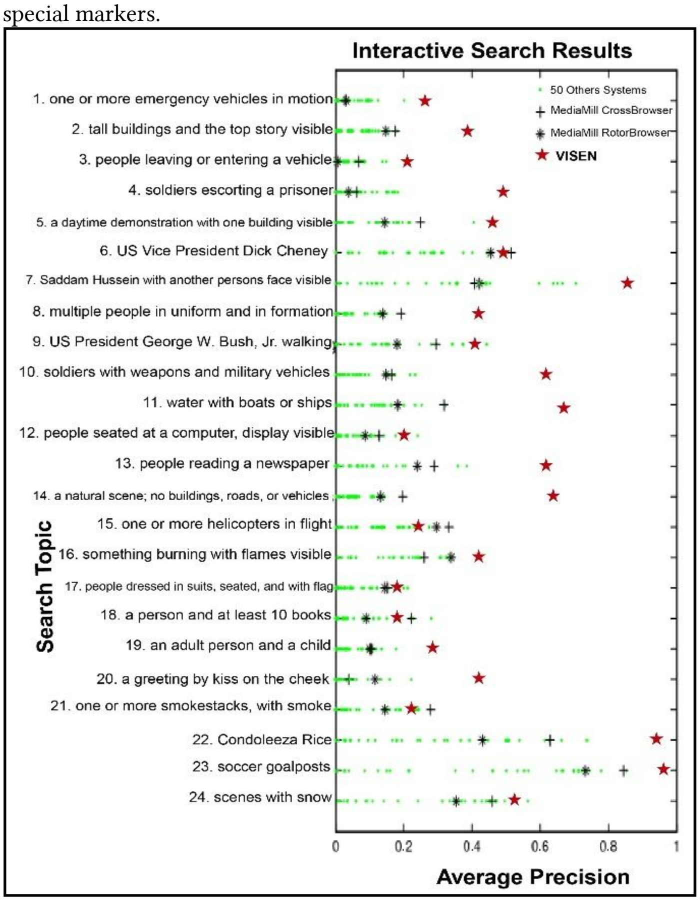


Figure 12: Comparison of VISEN

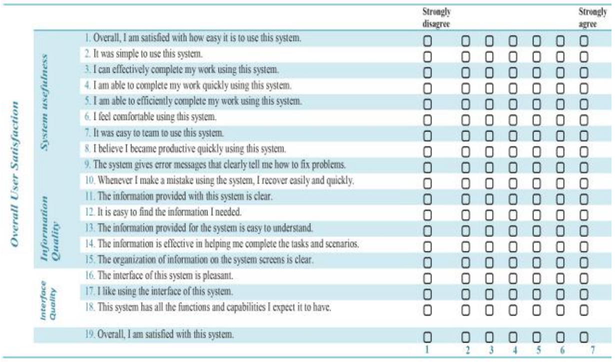
6.2.2. The user's analysis

Our proposed survey for the evaluation of our VISEN system is inspired by [34].

As it emerges from figure13, the first eight questions assess the system usefulness. Questions 9–15 assess the participants' satisfaction with the quality of the information associated with the system. Questions 16–18 provide a rating for the interface quality.

A user's study of 25 participants is performed to evaluate our system. The participants taking part in the experiments are students from computer Science University who are familiar with video search engines like "YouTube" and "Dailymotion". After the use of our system, the participants must fill a form containing the following questions.

The notes are comprised between 1 and 7: 1 'strongly disagree' and 7 'strongly agree'.



			Strongly disagree						Strongly agree
Overall User Satisfaction	System usefulness	1. Overall, I am satisfied with how easy it is to use this system.	□	□	□	□	□	□	□
		2. It was simple to use this system.	□	□	□	□	□	□	□
		3. I can effectively complete my work using this system.	□	□	□	□	□	□	□
		4. I am able to complete my work quickly using this system.	□	□	□	□	□	□	□
		5. I am able to efficiently complete my work using this system.	□	□	□	□	□	□	□
		6. I feel comfortable using this system.	□	□	□	□	□	□	□
		7. It was easy to team to use this system.	□	□	□	□	□	□	□
		8. I believe I became productive quickly using this system.	□	□	□	□	□	□	□
	Information Quality	9. The system gives error messages that clearly tell me how to fix problems.	□	□	□	□	□	□	□
		10. Whenever I make a mistake using the system, I recover easily and quickly.	□	□	□	□	□	□	□
		11. The information provided with this system is clear.	□	□	□	□	□	□	□
		12. It is easy to find the information I needed.	□	□	□	□	□	□	□
		13. The information provided for the system is easy to understand.	□	□	□	□	□	□	□
		14. The information is effective in helping me complete the tasks and scenarios.	□	□	□	□	□	□	□
		15. The organization of information on the system screens is clear.	□	□	□	□	□	□	□
	Interface Quality	16. The interface of this system is pleasant.	□	□	□	□	□	□	□
		17. I like using the interface of this system.	□	□	□	□	□	□	□
		18. This system has all the functions and capabilities I expect it to have.	□	□	□	□	□	□	□
		19. Overall, I am satisfied with this system.	□	□	□	□	□	□	□
			1	2	3	4	5	6	7

Figure 13: The Computer System Usability Questionnaire "CSUQ"

Fig.14 summarizes the CSUQ questionnaire participants' scores. Subsequently, we provide a graphical representation (in the form of a histogram) of these answers. An interpretation of this test finding is included at the end of this section. It should be recalled that we used a 1-7 scale with 7 being the best possible score and "1" the wrong answer.

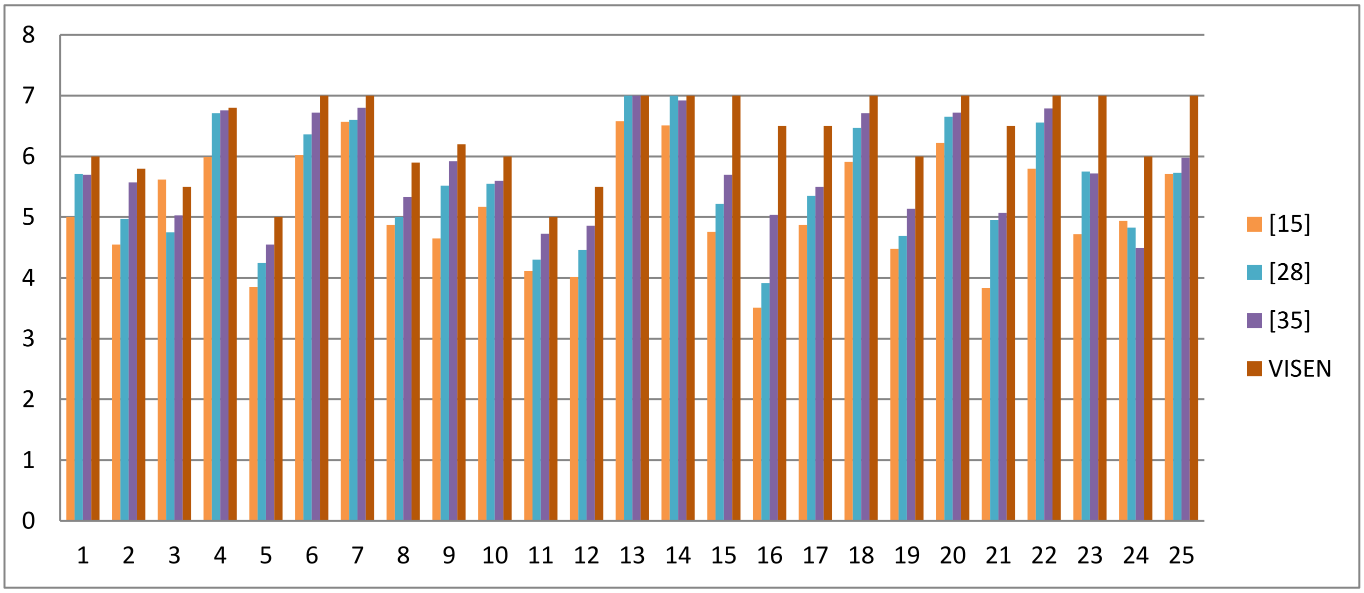


Figure 14: Comparison between the average score of [15], [28], [35] and our VISEN system of 25participants

The histogram (Fig.14) shows that the maximum votes were concentrated on the middle scores between 5 and 7. Based on the result of this experiment, it can be concluded that using our system does not entail any major drawback. Nevertheless, there is a need to improve some points, such as concepts similarity.

The 2nd step of the assessment, we will compare our system with some semantic retrieval systems (Fig.15). Since the score of our results is 85.85% of the overall user's satisfaction, 78.14% of System usefulness, 81.42% of information quality and 82.85% of interface quality, our VISEN scheme has the highest score among the other state-of-the-art systems. In fact, our proposed VISEN system outperforms those of Ben Hallima and Hamroun [13], U. Rashid [15] and M. Hamroun [28, 36]

Having analyzed the obtained test results, we can affirm that our video search system has proven reliable and efficient. These tests enabled us to value the performance of the formula used for concept weighting as well as the formula used for inter-concept similarity calculation. We may conclude that our system managed to achieve our goal to a certain extent.

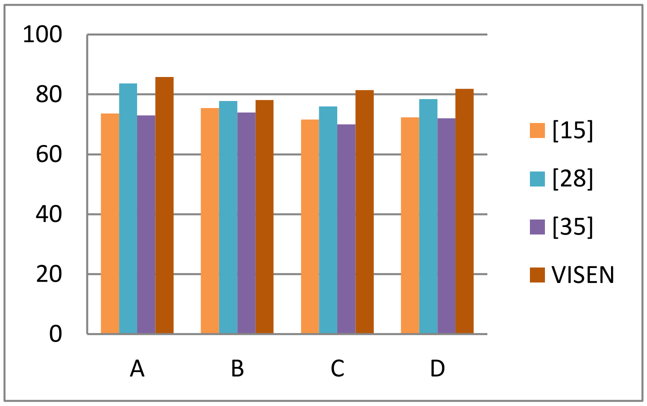


Figure 15: Comparison between the score value of [15], [28], [35] and our VISEN system

7 Conclusion

The general framework of our work, in this paper, is the automatic indexing of video based on its semantic content. In fact, we have proposed a semantic indexing model. Our contribution in this framework is based on several approaches, (i) DCM classification method for better indexing of content, (ii) new query expansion and relevance feedback methods putting the user at the centre of the retrieval process. We implemented a prototype entitled "VISEN" to validate our video semantic indexing models. The developed prototype enables users to easily access the desired video. To test our VISEN (video retrieval system) prototype, we used the audio-visual corpus (TRECVID2015), which is characterized by its size and the importance of its heterogenized content.
Our aim, as a first perspective, is to merge low-level descriptors and high-level in the retrieval process, i.e., which implies that the user can indicate the rate that a retrieval must be based on visual or/and semantic. The second part of our perspective is about considering special relationships between concepts. Indeed, for the moment, our automatic system detects the concepts without considering any relation between them. It will be interesting to create special relationships between the concepts or objects like with Belz et al.'s works [33]. For example, "*Singing*" and "*walking*" are concepts of the human actions. The concept of "*cycling*" is defined in TRECVID as "*a person riding a bicycle*". Although both a "*bicycle*" and a "*person*" exist in Fig.16, this image does not fit "*Bicycling*" because the person is not riding the bicycle. These indicators are important for some concepts, which indicates that not only the detected concepts are important but also their special relationship.

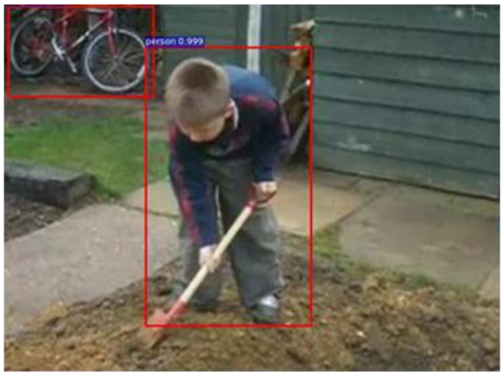


Figure 16: Example of VISEN error

REFERENCES

[1] I. Mbaye. Navigation conjointe dans une base de vidéos et d'images, 2006.

[2] S. Lefèvre, C. L'Orphelin and N. Vincent.Extraction multicritère de texte incrusté dans les séquences vidéo. Colloque International sur l'Ecrit et le Document (CIFED), 2004.

[3] C. Wolf and J. M. Jolion. Extraction de texte dans des vidéos : le cas de la binarisation. In Proceedings of RFIA, pages 145-152, 2000.

[4] M. Christel and A. Hauptmam. The use and utility of high-level semantic features. In CIVR, pages 134-144, 2005.

[5] C. G. M. Snoek, M. Worring, D. C. Koelma, and A. W. M. Smeulders. A learned lexicon-driven paradigm for interactive video retrieval. IEEE Transactions on Multimedia, pages 280-292, 2007.

[6] M. Worring, C. Snoek, O. de Rooji, G. P. Nguyen, R. Van Balen and D. Koelna. Médiamill : Advanced browsing in news vidéo archives. In CIVR, pages 533-536, 2006.

[7] V. Stefanos, M. Anastasia, K. Paul, D. Anastasios, M. Vasileios and K. Ioannis. VERGE : A Video Interactive Retriveal Engine, 2010.

[8] J., Tang, S., Yan,, R., Hong, G.J., Qi, T.S., Chua. Inferring semantic concepts from community-contributed images and noisy tags. In Proc. ACM Conference Multimed, pages 223– 232, 2009.

[9] S. Tang, Y.T. Zheng, Y. Wang, T.S. Chua, Sparse ensemble learning for concept detection. In IEEE Trans journal. Multimed, pages 43 – 54, 2012.

[10] M. Sjöberg, V. Viitaniemi, M. Koskela, J. Laaksonen. PicSOM Experiments. In TRECVID, 2009.

[11] J. Slimi, A. Ben Ammar, A.M. Alimi. Interactive video data visualization system based on semantic organization. In 11th International Workshop on Content-Based Multimedia Indexing, pages161-166, 2013.

[12] J. Slimi, S. Mansouri, A. Ben Ammar, A. M. Alimi. Video exploration tool based on semantic network. In OAIR, pages, 213-214, 2013.

[13] M. Ben Halima, M. Hamroun, S. Moussa and A.M. Alimi. An interactive engine for multilingual video browsing

using semantic content", International Graphonomics Society Conference IGS, Nara Japan, pages 183-186, 2013.

[14] LS. Kennedy, SF. Chang. A reranking approach for context-based concept fusion in video indexing and retrieval. In: Proceedings of the 6th ACM international conference on Image and video retrieval, CIVR'07, pages 333–340. ACM, New York, NY, USA, 2007.

[15] U. Rashid, M. Viviani, G. Pasi. A graph-based approach for visualizing and exploring a multimedia search result space. Inf. Sci.370-371, pages 303-322, 2016.

[16] Z. Zhang, W. Li, C. Gurrin, Alan F. Smeaton Faceted Navigation for Browsing Large Video Collection. MMM. pages 412-417. 2016

[17] M.S. Lew, N. Sebe, C. Dheraba. Content-based multimedia information retrieval: state of the art and challenges, J. Trans. Multimed Comput. Commun Appl 2, pages1– 19, 2016.

[18] A .W.M. Smeulders, M. Worring, S. Santini, A . Gupta, R. Jain. Content based image retrieval at the end of the early years. J. IEEE Trans. Pattern Anal. Mach. Intell. 22, pages 1349– 1380, 2000.

[19] D. Etter. KB Video Retrieval at TRECVID 2009. In TRECVID 2009.

[20] N. Elleuch, I. Feki, A. Ben Ammar, H. Karray, A. M. Alimi. REGIM at TRECVID2009: Semantic Access to Multimedia Data. In TRECVID 2009.

[21] N. Ellouze, N. Lammari, E. Métais, M.Ben Ahmed. CITOM: Approche de construction incrémentale d'une Topic Map multilingue. In Actes du Workshop RISE Recherche d'information sémantique (associé à INFORSID 2009), Catherine Roussey et Jean-Pierre Chevallet, pages, 65-85, 2009.

[22] L., Rossetto, I., Giangreco, C., Tanase and H., Schuldt. Multimodal Video Retrieval with the 2017 IMOTION System. In ICMR'17, June 6–9, 2017, Bucharest, Romania, 2017.

[23] C.G.M. Snoek, S. Cappallo, D. Fontijne, D. Julian, D.C. Koelma, P. Mettes. Qualcomm Research and University of Amsterdam at TRECVID 2015. Recognizing Concepts, Objects, and Events in Video, 2015.

[24] K. Ueki and T. Kobayashi. Waseda at TRECVID 2015: Semantic Indexing. TREVVID 2015.

[25] N. Elleuch, A. Ben Ammar and A.M. Alimi. A generic framework for semantic video indexing based on visual concepts/contexts detection. In Mutimedia Tools and application, 2015.

[26] P. Faudemay and C. Seyrat. Intelligent delivery of personalised video programmes from a video database. International workshop on Database anx EXpert systems Applications, pages 172-177, 1997.

[27] J.J. Racchio. Relevance Feedback in Information Retrieval. The Smart System Experiments in Automatic Document Processing, pages 313-323, 1971.

[28] M. Hamroun, S. Lajmi, H. Nicolas and I. Amous. ISE: Interactive Image Search Using Visual Content. In Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018) - Volume 1, pages 253-261, Madeira Portugal, 2018.

[29] R. Rada, H. Mili, E. Bichnell et M. Blettner. Development and application of a metric on semantic nets. In IEEE Transaction on Systems, Man, and Cybernetics, pages 17-30, 1989.

[30] Z. Wu and M. Palmer. Verb semantics and lexical selection. In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, pages 133- 138. 1994.

[31] P. Resnik. Using information content to evaluate semantic similarity in taxonomy. In Proceedings of 14th International Joint Conference on Artificial Intelligence, Montreal, 1995.

[32] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical Taxonomy. In Processing of international Conference onretrieval in Computational Linguistics, Taiwan 1997.

[33] Belz et al., Describing Spatial Relationships between Objects in Images in English and French (http://www.aclweb.org/anthology/W15-2816) 2015.

[34] J.R. Lewis. IBM computer usability satisfaction questionnaires:psychometric evaluation and instructions for use, Int. J. Hum.-Comput, pages, 57–78, 1995.

[35] M. Hamroun, S. Lajmi, H. Nicolas and I. Amous. An Interactive Video Browsing With VINAS System. In Proceedings of the 15th ACS/IEEE International Conference on Computer Systems and Applications, pages 1-8, 2018.

Data Mining Ancient Scripts to Investigate their Relationships and Origins

Shruti Daggumati
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
sdagguma@cse.unl.edu

Peter Z. Revesz
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
revesz@cse.unl.edu

ABSTRACT

This paper describes a data mining study of a set of ancient scripts in order to discover their relationships, including their possible common origin from a single root script. The data mining uses convolutional neural networks and support vector machines to find the degree of visual similarity between pairs of symbols in eight different ancient scripts. Among the surprising results of the data mining are the following: (1) the Indus Valley Script is visually closest to Sumerian pictographs, and (2) the Linear B script is visually closest to the Cretan Hieroglyphic script.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning*; *Machine learning approaches*; • **Information Systems** → *Data mining*; • **Human-centered computing** → *Visualization*; *Treemaps.*

KEYWORDS

Convolutional Neural Networks, Data Analytics, Data Mining, Indus Valley Script, Sumerian Pictographs, Support Vector Machines, Data Visualization, Machine Learning

ACM Reference Format:
Shruti Daggumati and Peter Z. Revesz. 2019. Data Mining Ancient Scripts to Investigate their Relationships and Origins. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331076.3331116

1 INTRODUCTION

The data mining work in this paper is motivated to help decipher ancient scripts such as the still undeciphered Indus Valley Script [24]. The idea is that if an undeciphered script can be matched with an already deciphered script, then the phonetic values of the symbols in the deciphered script can be reasonably expected to match the phonetic values of the corresponding symbols in the undeciphered script.

We applied various data mining methods to compare and analyze the relationship among the following ancient scripts: Brahmi, Cretan Hieroglyphs, Greek, Indus Valley, Linear B, Phoenician, Proto-Elamite, and Sumerian Pictographs. Our data mining yields a script family tree with a common origin of all these scripts. A particularly interesting finding of our data mining is that the Indus Valley Script seems to derive from the Sumerian Pictographs. Our finding is supported by the following observations of other authors. First, it is known that intensive trade existed, mainly by sea between the ancient civilizations in Mesopotamia and the Indus valley, and the urbanization, irrigation technology, social organization, commercial patterns, and numerous other features of the Indus Valley civilization bears a close resemblance to the Sumerian model [4, 9]. Second, the ancient Sumerian records referred to the Indus Valley Civilization as *Meluhha*, which means "high country" in Dravidian languages according to Parpola [24] and may be related to the present day region of *Baluchistan*.

The rest of this paper is organized as follows. Section 2 describes the dataset of the ancient scripts and texts which we used as a data source. Section 3 describes the machine learning methodologies that we used for the computerized comparison of the visual characteristics of pairs of symbols from the different scripts. Section 4 presents the experiments and results and analyzes the findings. Section 5 discusses related work. Finally, Section 6 gives some conclusions and directions for further research.

2 DATASET

In this section, we provide the historical background for all the scripts used in this work. We also describe how the datasets were created for the computations.

2.1 Brief Review if the Eight Scripts Considered

2.1.1 Brahmi. Brahmi is the second oldest South Asian script, after the Indus Valley Script. The Brahmi script is an abugida, which uses a system of diacritical marks to denote vowel association with the consonant symbols. The direction of writing for the Brahmi script is left to right. Much like the Indus Valley Script, the Brahmi script has a debated origin.

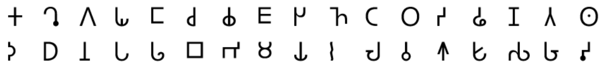


Figure 1: Sample Brahmi script symbols.

2.1.2 Cretan Hieroglyphs. Cretan Hieroglyphs was the first writing of the Minoans and predecessor to Linear A, which in turn gave rise to Linear B and Cypriot. It was used between 2100 to 1700 BC [2, 23]. The second author proposed recently a decipherment of Cretan Hieroglyphs [36], but there are many alternative proposals.

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331116

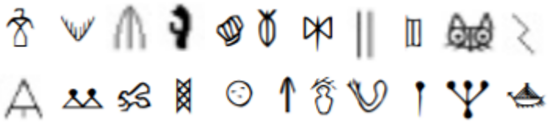


Figure 2: Sample Cretan Hieroglyphs.

2.1.3 Greek. There were many variants of the early Greek alphabet, each suited to a local dialect. Eventually, the Ionian alphabet was adopted in all Greek-speaking states. Ancient Greek is a full (consonants and vowels) alphabet. Greek was written from around 800 BC to the 5th century in both a right-to-left and a boustrophedonic style, but later it transitioned to a left-to-right writing system [5].

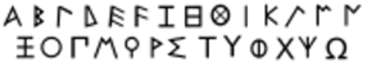


Figure 3: The 26 letters of the ancient Greek alphabet.

2.1.4 Indus Valley. The Indus Valley Script is an undeciphered script, which was used between 2400 and 1900 BCE [25]. It is stated to be a logographic and syllabic writing system, written from right to left [25].

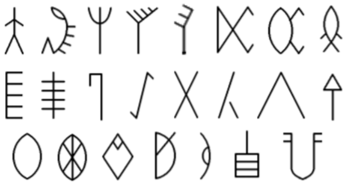


Figure 4: Sample Indus Valley script symbols.

2.1.5 Linear B. Linear B was used in Mycenaean Greece and is the oldest known Greek writing [15]. Linear B remained a mystery until 1952 when Michael Ventris deciphered Linear B showing that it is an archaic version of Greek [3]. Linear B is a syllabic writing system where in general each syllable begins with a single consonant, which is followed by a single vowel.

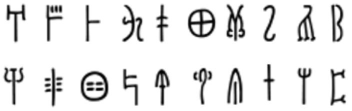


Figure 5: Sample Linear B symbols.

2.1.6 Phoenician. The Phoenician alphabet was used from 1200 to 150 BC in the eastern Mediterranean [13]. The Phoenician alphabet is an abjad (only consonants with no vowels) writing system, written from right to left, which consists of 22 letters representing consonants [13]. The Phoenician alphabet may derived from Egyptian Hieroglyphs [16] or Linear B [35].

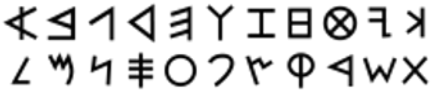


Figure 6: The 22 letters of the Phoenician alphabet.

2.1.7 Proto-Elamite. The Proto-Elamite script was briefly used between the end of 4000 to the beginning of 3000 BCE in present-day Iran and southern Iraq [11]. The script uses around 1900 non-numerical signs, although 1700 of those signs only appear a maximum of nine times in the 1600 Proto-Elamite texts [8]. The Proto-Elamite script is said to be logographic or ideographic [11] and is also considered undeciphered.

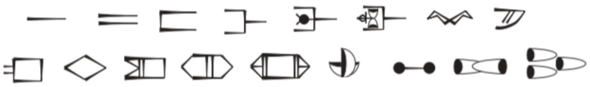


Figure 7: Sample Proto-Elamite script symbols.

2.1.8 Sumerian Pictographs. The Sumerian language is distantly related to both the Uralic and the Dravidian language families [28, 41]. However, the Sumerian Pictographs are considered an independent development by most researchers [11]. The Sumerian pictographic script is primarily a syllabic and logographic writing system. It was written from left to right, and it and its cuneiform descendant were used from 3100 BCE to 1st century AD [11].

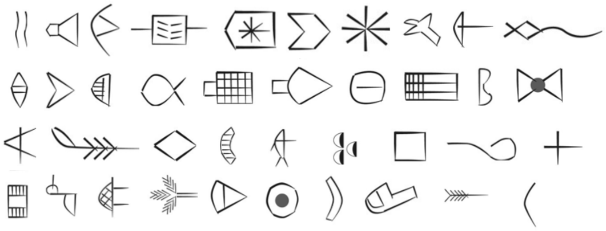


Figure 8: Sample Sumerian Pictographs.

2.2 Data Source

The eight different scripts outlined in the previous section were used as a data source. For the Brahmi script we use 34 of the symbols (Figure 1), for the Cretan Hieroglyphs we use 22 symbols (Figure 2), for Greek we use all 27 symbols (Figure 3). For the Indus Valley Script, we use 23 symbols (Figure 4) which were symbols with the highest frequencies because the Indus Valley Script has at least over 400 symbols and symbols that occur only once or twice are likely to be insignificant [44]. For Linear B we use 20 symbols (Figure 5), for the Phoenician alphabet we use all 22 symbols (Figure 6), for the Proto-Elamite script we use 17 symbols (Figure 7), and for the Sumerian Pictographs we use 34 symbols (Figure 8).

2.3 Data Gathering and Processing

Our dataset is modeled after the MNIST image database [21]. Each symbol in our dataset has 780 training images and 120 validation images, that is a total of nine hundred images associated with each symbol. The images used were hand generated and computer modified via minor skewing and distortion. Each image is 50x50 pixels, grayscale, and centered in the 50x50 region using the center of mass. These features are the necessary preprocessing steps for each dataset.

3 SOFTWARE ARCHITECTURE

3.1 Convolutional Neural Network (CNN)

We created neural networks using Python and TensorFlow with a Keras wrapper. The constructed neural networks have various levels of accuracy, depending on the script learned. The architecture of our convolutional neural network is similar to the LeNet model [20] with a modification on the output classification as shown in Figure 9. The main deviance from the original LeNet model is that we use an SVM classifier for the final dense layer instead of a Softmax layer. Previous works have shown this to be useful in recognition of other languages [10] or even when the sample set is more than ten [21].

Starting with our image size of 50x50 we first apply a convolution using a filter size of 5x5, which reduces our image to 46x46. After this we apply a pooling layer which reduces our image size by half, entailing a 23x23 image. We then add one more convolution layer using a 4x4 filter which reduces our image size to 20x20. Then we apply a pooling layer which reduces our image in half again to 10x10. We then pass the image to a fully connected flattened layer of 1024 neurons, which then passes the data to our SVM (see Section 3.2). Each convolution layer has a Rectified Linear Unit (ReLU) activation function. ReLU is often used as the activation function of choice for most CNN architectures. The ReLU activation function produces zero as an output when $x \leq 0$ or it produces a linear value with slope of one when $x > 0$. Each pooling layer employs max pooling. Each 2x2 filter takes the maximum value of the four quadrants to use for the feature map. To combat overfitting we use a drop rate of 0.4. Each CNN uses the Adam optimizer with learning rate of 0.001. Adam is an adaptive learning rate optimization algorithm that was designed specifically for training deep neural networks [19].

3.2 Support Vector Machine (SVM)

The generated SVM is implemented in Python and uses Python library packages. SVMs were designed for binary classification. However, in our research, we use SVMs for a multiclass problem. Generally, for classification problems in CNNs, the last layer uses Softmax. In this research, we use L2-SVM which is differentiable and optimizes the sum of the squared errors. The L2-SVM also minimizes the squared hinge loss. The optimization function for the L2-SVM is shown below, where w is an N-dimensional weight vector, b is the bias terms, and ξ_i are slack variables, and C is the penalty parameter.

Minimize:

$$\frac{1}{2}\,\|w\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \tag{1}$$

Subject to:

$$y_i(x_i \cdot w + b) \geq 1 - \xi_i \qquad i = 1, ..., N \tag{2}$$

As mentioned previously, training the classifier using the L2-SVM objective function outperforms other methods such as L1-SVM or Softmax regression [48].

3.3 Prediction Classifier

In addition to creating a CNN+SVM classifier per each script, we also look at the similarities between two pairs of scripts. The trained CNN+SVM model for every script is passed into the other seven script models. The basic idea of the predictive classifier is illustrated in Figure 10.

Each similarity matrix produced by the CNN+SVM for the eight scripts has different NxM dimensions based on the number of symbols in each script. We create the following two measures to see the strength between two scripts:

(1) The **Average of All** takes the average of the strongest probability matches for each symbol pair.
The rational is that taking the average of the strongest matches between two scripts takes into account all the symbols in each script. If a symbol provided as input has a low correlation with all of the trained symbols, the overall average would reflect this.

(2) The **Selective Average** only considers pairs of symbols which have higher than seventy-five percent similarity match and then take the average.
The rational is that the selective average provides two measures in regards to the similarity of two scripts. It provides not only a higher overall average in comparison to taking the average overall but also the number of symbols which are the closest together. The selective average also takes into account that a script may not completely stem from only one script. Therefore not all symbols may have a high correlation.

3.4 Classification Trees

Each CNN+SVM for the prediction classifier has seven mappings for the eight scripts. The strength between the scripts is provided using the two averages presented in Section 3.3. In addition, we take into account the number of symbols between the scripts which have a correlation value ≥ 75%.

To create a classification tree we employ two different algorithms for the two measures as listed below.

(1) **Similarity:** *The scripts which have a higher correlation are paired.* We use WPGMA (Weighted Pair Group Method with Arithmetic Mean) to create our dendrogram for the scripts. The WPGMA algorithm creates a dendrogram that displays the structure in the similarity matrix. The nearest two clusters are combined at each step i.e. clusters x and y are combined to create $x \cup y$. Then the distance to another cluster z is the mean of the distances between z and $x \cup y$ as shown in Equation (3). Since we use a similarity matrix as input to the WPGMA method, we use the complement of the matrix. That is, now the smaller values indicate higher similarity.

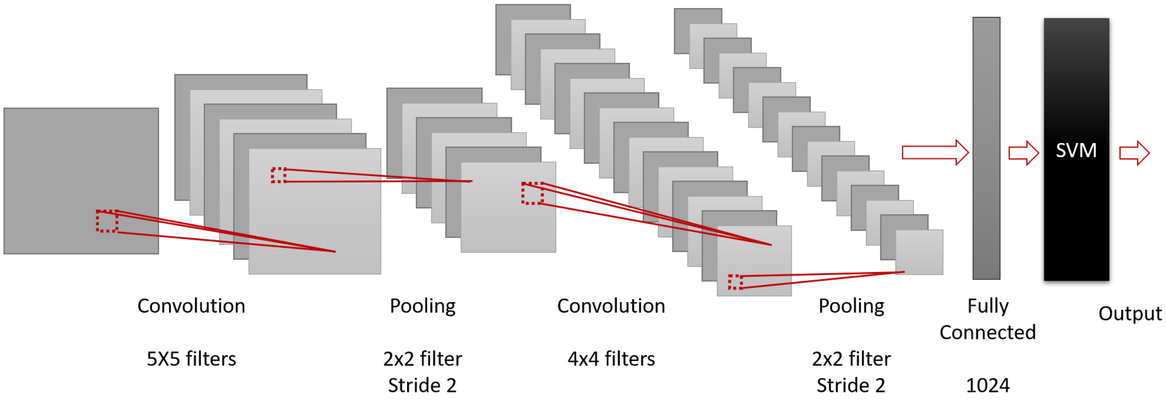


Figure 9: The architecture of our CNN+SVM classifier.

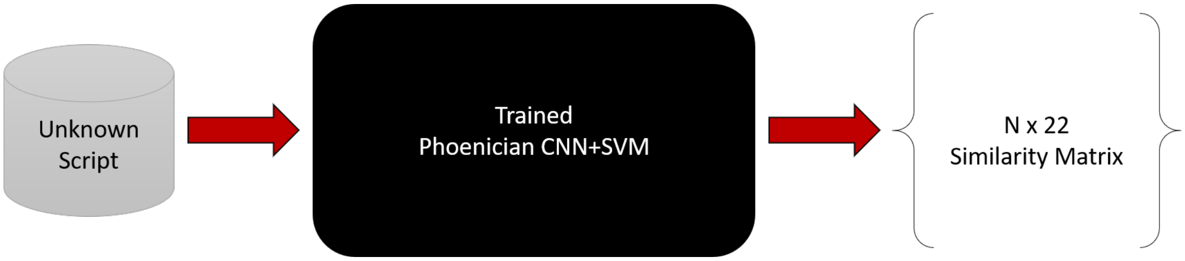


Figure 10: The predictive CNN+SVM classifier comparing the other seven scripts to the Phoenician alphabet. The unknown script is replaced with any of the seven other scripts. The size of the matrix is dependent on the number of symbols in the unknown script provided.

$$d_{(x\cup y),z} = \frac{d_{x,z} + d_{y,z}}{2} \quad (3)$$

(2) **Hierarchical:** *The scripts are ancestor/descendant of another script.* The hierarchical tree generation is implemented again using WPGMA but also considering the time period when each script was used. By doing this we can create a descendant tree, which highlights the possible descendant of each script. The details are shown below in Algorithm 1.

Algorithm 1 Time-Based Descendant Tree

```
Create parent node P
Create a node for each script
for all Closest Script Pairs $S_x$ and $S_y$ do
    if $S_x.Time > S_y.Time$ then
        Parent of $S_x$ is P
        Parent of $S_y$ is $S_x$
    else
        Parent of $S_y$ is P
        Parent of $S_x$ is $S_y$
for all Singleton Scripts $S_z$ do
    Parent of $S_z$ is P
return Tree
```

4 EXPERIMENTAL RESULTS AND ANALYSIS

In this paper, we have three main fundamental building blocks: the dataset creation, the CNN+SVM classifier, and the hierarchical tree creator. The latter two portions were first independently verified and then combined to create a final product.

4.1 Validation of the Script Classifier

Each script has its own CNN+SVM classifier. The accuracy of the different scripts is shown in Table 1 with an increase of epochs (step size = 25). We see that for all the scripts at 25 epochs we have already reached the 90% accuracy, similarly to MNIST CNNs.

4.1.1 Script Prediction. For each script, its CNN+SVM classifier has an almost perfect accuracy at 100 epochs. Due to that, we see whether the CNN+SVM can be used to find ancestors and/or descendants of other scripts. We partition this experiment into two categories: **Known Origin** and **Unknown Origin** The known origin scripts validate our framework and ensure that our tool is capable of reproducing established results. Some specific categorizations:

(1) **Known Origin:** Phoenician is the ancestor of ancient Greek, as mentioned already by Herodotus, and Brahmi, via Aramaic. Cretan Hieroglyphic script an ancestor of Linear B.
(2) **Unknown Origin:** The Sumerian Pictographs, the Indus Valley, and the Proto-Elamite scripts have unknown ancestors and descendants.

Table 1: Validation Accuracy

	Number of Epochs			
	25	50	75	100
Brahmi	95.09	98.15	98.24	99.35
Cretan Hieroglyphs	91.09	92.84	94.47	97.53
Greek	93.49	96.26	97.23	98.63
Indus Valley	93.50	95.70	96.85	98.23
Linear B	91.19	93.15	96.42	99.48
Phoenician	93.18	94.77	95.36	97.52
Proto-Elamite	91.93	94.55	97.05	99.09
Sumerian Pictographs	90.79	93.21	96.94	97.40

4.1.2 Validation of Our Method - Known Script Prediction. By using the prediction techniques we aim to see the similarities between the scripts. We first validate our thoughts by passing Greek into the trained Phoenician CNN-SVM and vice-versa. Similarly, we repeat this experiment with Linear B and the Cretan Hieroglyphs.

As seen in Figures 11 and 12, the heatmaps of the similarity matrices between Phoenician and Greek indicates high correlation on the diagonal. This indicates the Phoenician and Greek have an almost one-to-one mapping. We see that this result is validated by the known mapping between Greek to Phoenician as shown in Table 2. We find similar results with Linear B and Cretan Hieroglyphs, which also indicates that the Cretan Hieroglyphs and Linear B have an almost one-to-one mapping.

4.1.3 Unknown Origin Script Prediction. Since the CNN+SVM predictor worked well on the known origin scripts, yielding the expected ancestor-descendant relationships, we can safely use it for the unknown origin scripts too. As visualized in Figure 13, the Sumerian Pictographs and the Indus Valley script have a fairly strong correlation and an almost one-to-one mapping similar to the relation between Phoenician and Greek and between Cretan Hieroglyphs and Linear B. Table 3 notes the number of symbols which have a ≥ 75% correlation between scripts.

4.2 Tree Visualization Analysis

The similarity matrices shown in the previous sections produce the classification and hierarchy trees as shown in Figures 14 and 15, respectively.

4.2.1 Classification Tree. Beside confirming the known origins noted earlier, the classification tree generated some interesting new results. In particular, Brahmi is closest to Phoenician and Greek. The visualization also shows that Brahmi, the Cretan Hieroglyphs, Greek, and Linear B and Phoenician form one branch of the classification tree, while Sumerian Pictographs are closest related to the Indus Valley script.

4.2.2 Hierarchy Tree. The hierarchical tree not only shows the similarity between two pairs of scripts but also visualizes that Greek is a descendant of Phoenician and Linear B is a descendant of Cretan Hieroglyphs. In addition, the Indus Valley script has been classified as a possible descendent of the Sumerian Pictographs. Brahmi and Proto-Elamite have an unknown ancestor. However, they have some similarities to the other scripts to assume an unknown hypothetical common origin of these eight scripts.

Table 2: Mapping between Greek and Phoenician.

Phoenician		Greek	
𐤀	aleph	Α	alpha
𐤁	beth	Β	beta
𐤂	giml	Γ	gamma
𐤃	daleth	Δ	delta
𐤄	he	Ε	epsilon
𐤅	waw	Ϝ or Υ	digamma or upsilon
𐤆	zayin	Ζ	zeta
𐤇	heth	Η	eta
𐤈	teth	Θ	theta
𐤉	yodh	Ι	iota
𐤊	kaph	Κ	kappa
𐤋	lamedh	Λ	lambda
𐤌	mem	Μ	mu
𐤍	nun	Ν	nu
𐤎	samekh	Ξ	xi
𐤏	ayin	Ο	omicron
𐤐	pe	Π	pi
𐤑	sade	Ϻ	san
𐤒	qoph	Ϙ	koppa
𐤓	res	Ρ	rho
𐤔	sin	Σ	sigma
𐤕	taw	Τ	tau
-		Φ	phi
-		Χ	chi
-		Ψ	psi
-		Ω	omega

5 RELATED WORK

5.1 Background - Indus Valley Script

Sir Alexander Cunningham, one of the first to encounter the Indus Valley script, assumed that the seals were foreign import. He later stated that Brahmi might be a descendant of the Indus Valley script.

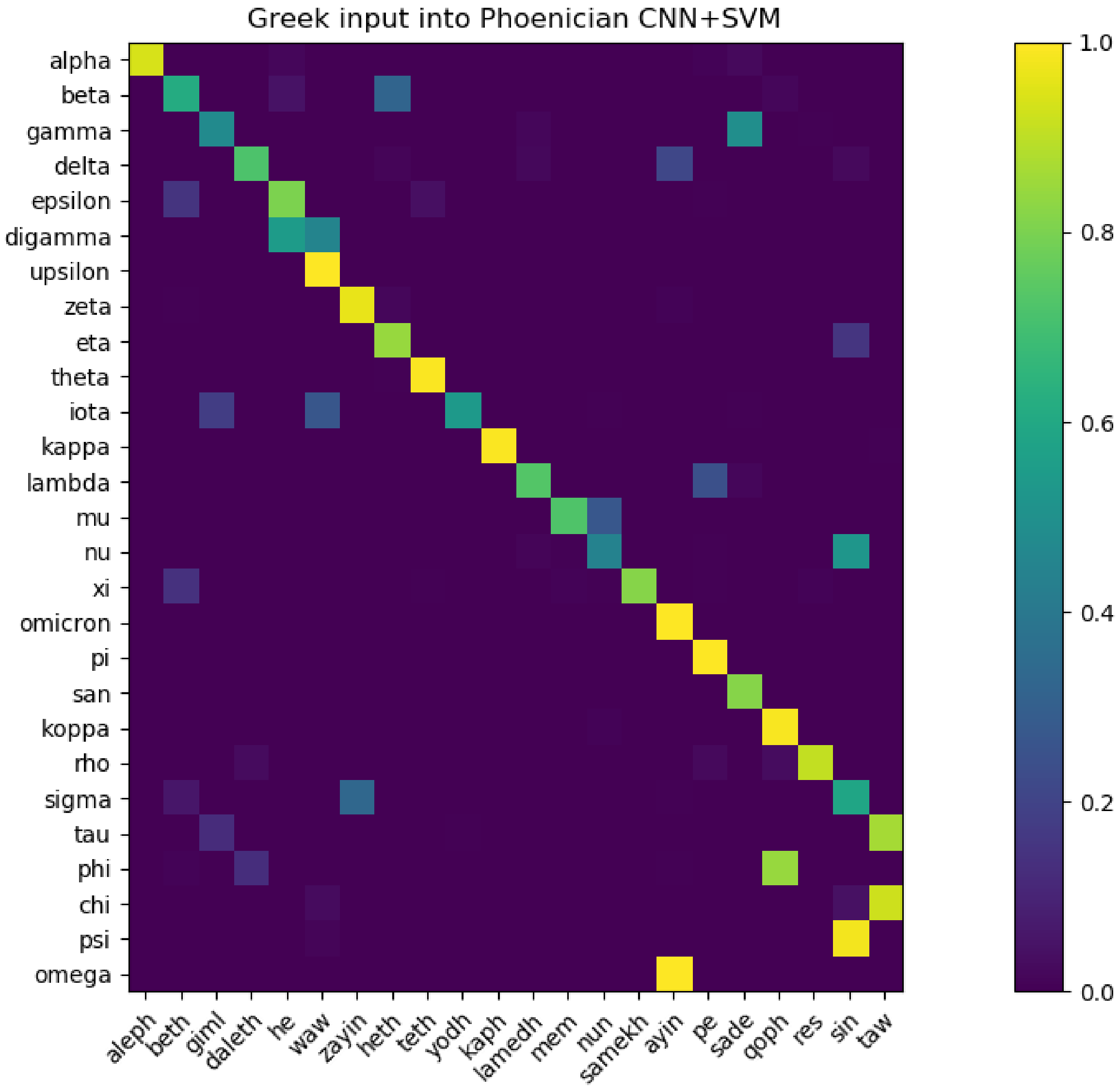


Figure 11: The Greek letters are provided as input to the trained Phoenician CNN+SVM.

Table 3: The number of symbols with correlation ≥ 75% between each pair of the eight scripts.

	Brahmi	**Cretan Hier.**	**Greek**	**Indus Valley**	**Linear B**	**Phoenician**	**Proto-Elamite**	**Sumerian Pict.**
Brahmi	34	-	-	-	-	-	-	-
Cretan Hier.	2	22	-	-	-	-	-	-
Greek	9	4	26	-	-	-	-	-
Indus Valley	8	5	9	23	-	-	-	-
Linear B	3	20	7	4	20	-	-	-
Phoenician	9	6	22	9	9	22	-	-
Proto-Elamite	2	2	2	4	0	3	17	-
Sumerian Pict.	6	6	7	20	5	7	3	39

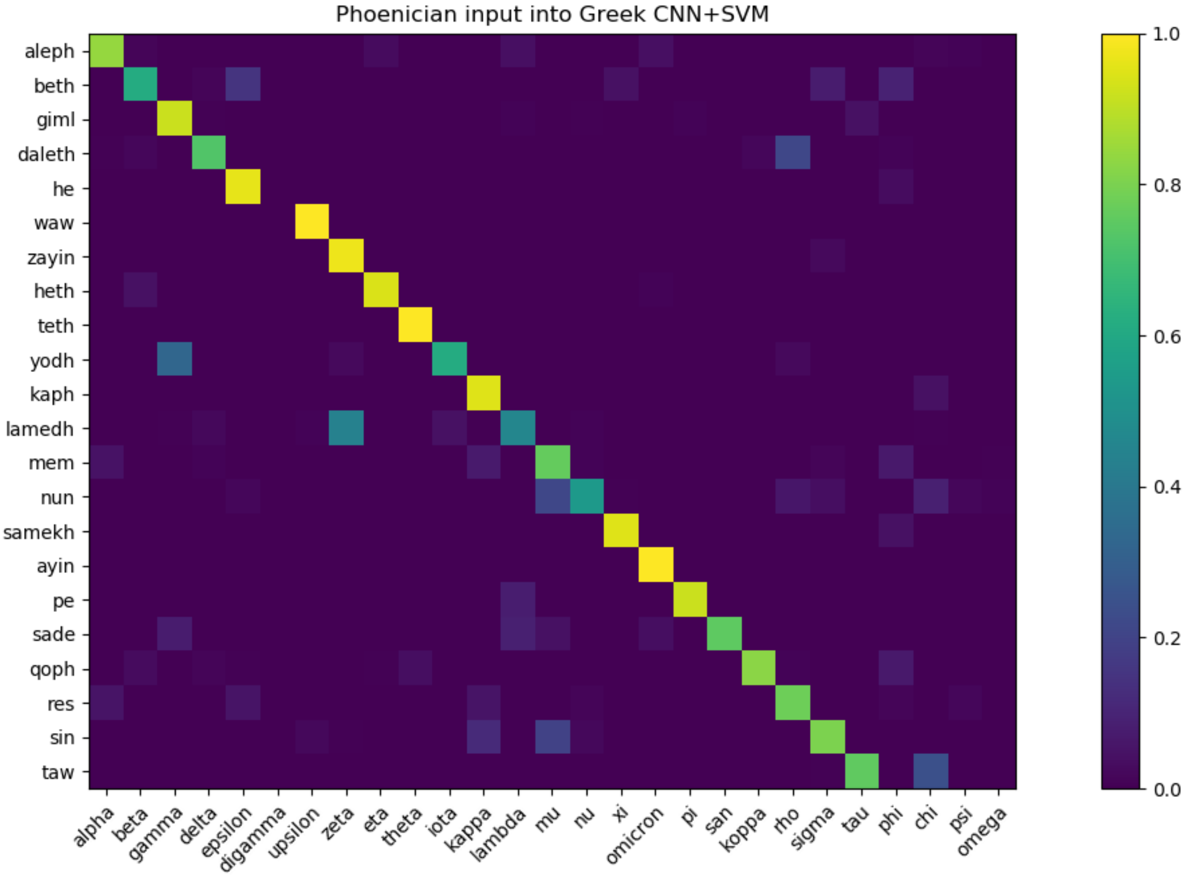


Figure 12: The Phoenician letters are provided as input to the trained Greek CNN+SVM.

Many other scholars have connected the Indus Valley script to Brahmi [29–31]. Many scholars also suppose that the Indus Valley Script expresses some Dravidian language [24, 26, 27, 43, 45, 46, 49], where the work from [49] was one of the first publications using computer aid to analyze the Indus Valley script.

Some scholars, such as McAlpin [22], support the Elamo-Dravidian hypothesis, which links the Dravidian to the Elamite languages. McAlpin also believes that the Indus Valley script could be part of the Elamo-Dravidian language family. That hypothesis is supported by evidence of extensive trade between Elam and the Indus Valley civilization.

There are a few scholars who believe that the Indus Valley script is not a language [12]. These scholars say that the Indus Valley script is comparable to nonlinguistic signs which symbolize family or clan names/symbols and religious figures/concepts. Regardless of it being a language or not, its similarity to the other scripts still suggests that the symbols were derived from Sumerian Pictographs.

Nevertheless, the brevity of Indus texts may indeed suggest that it represented only limited aspects of an Indus language. That is true of the earliest, proto-cuneiform, writing on clay tablets from Mesopotamia, around 3300 BC, where the symbols record only calculations with various products (such as barley) and the names of officials.

5.2 Machine Learning

Scholars have used various machine learning techniques to analyze and classify images and read text [17, 18].

Support vector machines and neural networks have been used to recognize a multitude of scripts. Artificial neural networks and SVMs were compared on the Devanagari script, a descendant of the Brahmi script [1]. Arabic handwritten recognition was recently studied using the CNN+SVM combination [10]. In addition, handwritten Chinese characters were analyzed using CNNs [14, 47]. Earlier work of the authors shows the similarity between the Indus Valley script and other scripts using CNNs [6, 7]. However, the use of neural networks to generate script families is a new domain.

5.3 Classification Trees

Revesz [34, 35] used hypothetical evolutionary tree reconstruction algorithms to analyze the development of the Cretan Script Family.

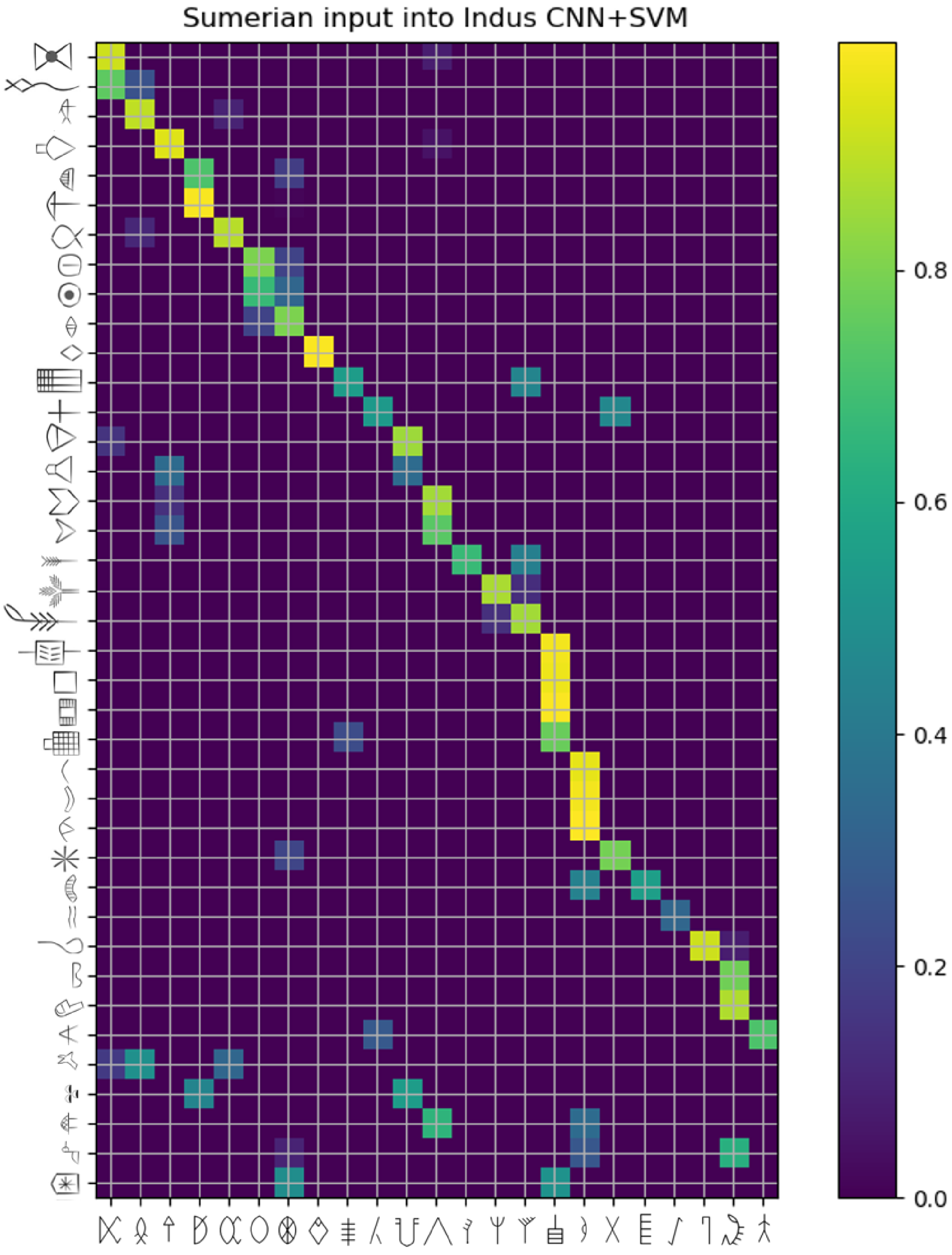


Figure 13: The Sumerian Pictograms are provided as input to the trained Indus Valley CNN+SVM.

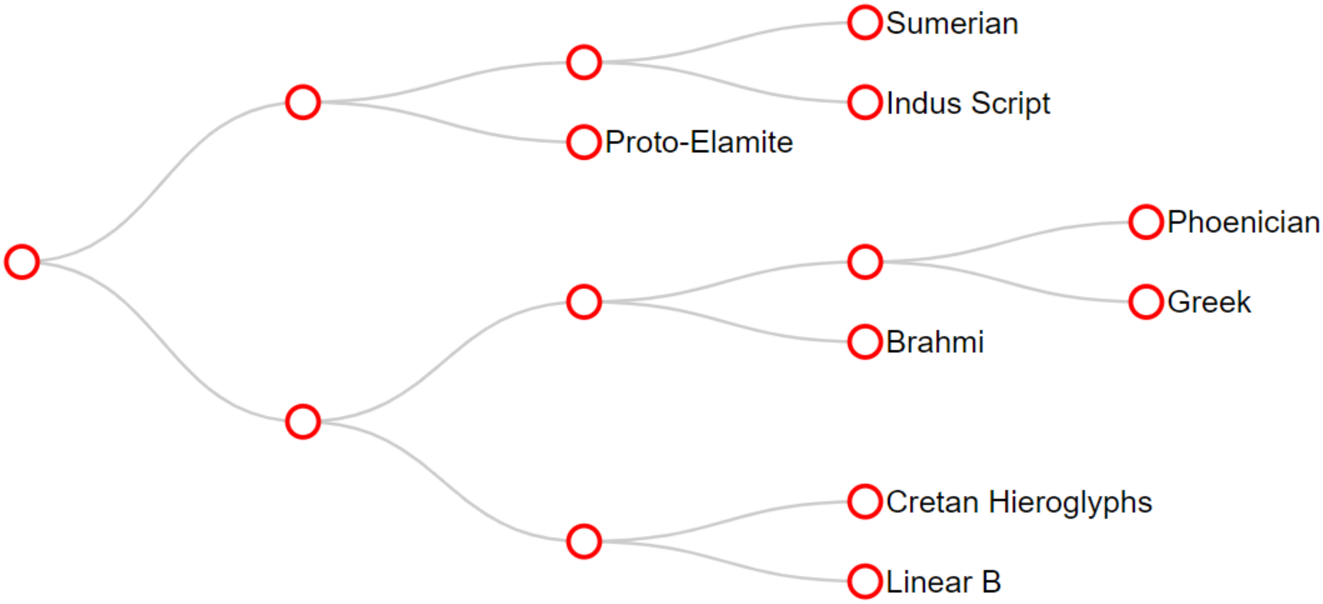


Figure 14: The classification tree created from the similarity matrix using WPGMA.

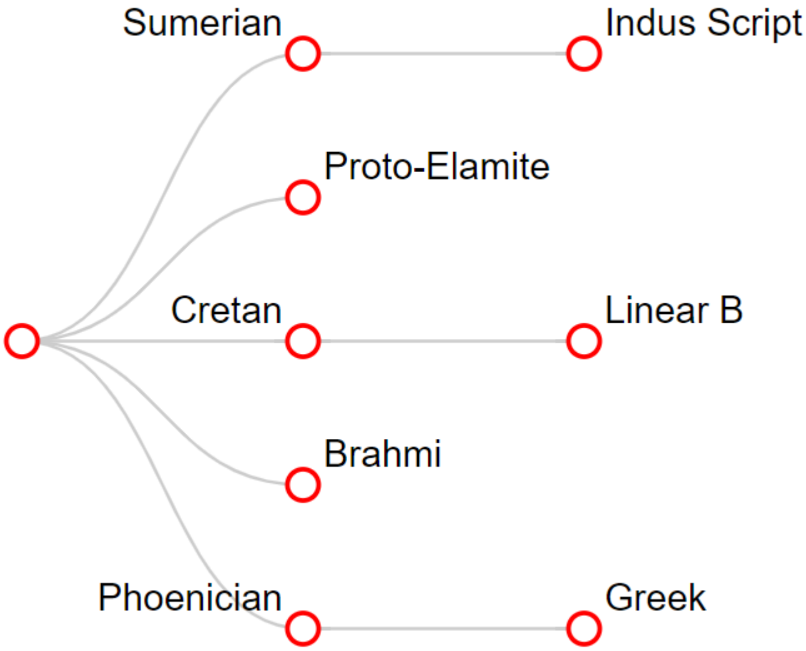


Figure 15: The hierarchical tree created by taking time into account.

The matching of Minoan Cretan Hieroglyphic and Linear A symbols with the Carian and the Old Hungarian alphabets yielded new phonetic values for the Cretan Hieroglyphic and Linear A symbols. The new phonetic values allowed the decipherment of the Linear A script [39], and the Cretan Hieroglyphic script [36], including the Arkalochori Axe [40] and the Phaistos Disk [37] inscriptions. The AIDA system [42] is an online Minoan inscriptions database that also contains some of these translations.

The origin of languages and scripts have long been studied by linguists. The use of genetic information tying civilizations and their languages have only recently been studied [32, 33, 38]. Using human archaeogenetics may provide new insight into the diffusion of human populations in association with various language families.

6 CONCLUSIONS AND FUTURE WORK

The invention and spread of writing was a giant step for humanity that is still largely shrouded in mystery. However, our data mining of ancient script databases revealed several interesting hitherto unknown relationships among the eight scripts studied. This work is only the beginning of a systematic neural networks-based exploration of an ancient script family that likely encompasses not only the eight scripts that we studied but also many others. Hence as a future work, we plan to add to our database other ancient scripts from the region of the Near East and the Mediterranean Sea. By adding more scripts to our CNN+SVM predictor system, we can obtain a more complete tree of visual similarities and reduce the remaining uncertainties in the development of one of the oldest script families in the world.

REFERENCES

[1] S. Arora, D. Bhattacharjee, M. Nasipuri, L. Malik, M. Kundu, and D. K. Basu. Performance comparison of SVM and ANN for handwritten Devnagari character recognition. *arXiv preprint arXiv:1006.5902*, 2010.

[2] J. G. P. Best and F. C. Woudhuizen, editors. *Ancient Scripts from Crete and Cyprus*, volume 9. Bill Archive, 1988.

[3] J. Chadwick. *The Decipherment of Linear B.* Cambridge University Press, 1958.

[4] D. Collon. Mesopotamia and the Indus: The evidence of the seals. In *The Indian Ocean in Antiquity*, pages 209–225. The British Museum and Kegan Paul International London/New York, 1996.

[5] B. F. Cook. *Greek Inscriptions*, volume 5. University of California Press, 1987.

[6] S. Daggumati. Similarity queries on script image databases. In A. Benczúr, B. Thalheim, T. Horváth, S. Chiusano, T. Cerquitelli, C. I. Sidló, and P. Z. Revesz, editors, *New Trends in Databases and Information Systems - ADBIS 2018 Short Papers and Workshops*, pages 391–401. Springer, 2018.

[7] S. Daggumati and P. Z. Revesz. Data mining ancient script image data using convolutional neural networks. In *Proceedings of the 22nd International Database Engineering and Applications Symposium*, pages 267–272. ACM, 2018.

[8] J. L. Dahl. Complex graphemes in Proto-Elamite. *Cuneiform Digital Library Journal*, 2005(6), 2005.

[9] C. Elisabeth and D. Caspers. Sumer, coastal Arabia and the Indus Valley in protoliterate and early dynastic eras: Supporting evidence for a cultural linkage. *Journal of the Economic and Social History of the Orient/Journal de l'histoire economique et sociale de l'Orient*, pages 121–135, 1979.

[10] M. Elleuch, N. Tagougui, and M. Kherallah. A novel architecture of CNN based on SVM classifier for recognizing Arabic handwritten script. *International Journal of Intelligent Systems Technologies and Applications*, 15(4):323–340, 2016.

[11] R. K. Englund. The Proto-Elamite script. In P. T. Daniels and W. Bright, editors, *The World's Writing Systems*, pages 160–164. Oxford University Press, 1996.

[12] S. Farmer, R. Sproat, and M. Witzel. The collapse of the Indus-script thesis: The myth of a literate Harappan civilization. *Electronic Journal of Vedic Studies*, 11(2):19–57, 2016.

[13] S. R. Fischer. *History of Writing.* Reaktion Books, 2004.

[14] M. He, S. Zhang, H. Mao, and L. Jin. Recognition confidence analysis of handwritten Chinese character with CNN. In *Proceedings of the 13th International Conference on Document Analysis and Recognition*, pages 61–65. IEEE, 2015.

[15] J. T. Hooker and J. H. Betts. *Linear B: An Introduction.* Bristol Classical Press, Bristol, UK, 1980.

[16] M. C. Howard. *Transnationalism in Ancient and Medieval Societies: The Role of Cross-Border Trade and Travel.* McFarland, 2014.

[17] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016.

[18] K. Jung, K. I. Kim, and A. K. Jain. Text information extraction in images and video: A survey. *Pattern Recognition*, 37(5):977–997, 2004.

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[20] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] Y. LeCun, C. Cortes, and C. Burges. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 1998.

[22] D. W. McAlpin. Proto-Elamo-Dravidian: The evidence and its implications. *Transactions of the American Philosophical Society*, 71(3):1–155, 1981.

[23] J.-P. Olivier. Cretan writing in the second millennium BC. *World Archaeology*, 17(3):377–389, 1986.

[24] A. Parpola. The Indus script: A challenging puzzle. *World Archaeology*, 17(3):399–419, 1986.

[25] A. Parpola. The Indus Script. In P. T. Daniels and W. Bright, editors, *The World's Writing Systems*, pages 165–171. Oxford University Press, 1996.

[26] A. Parpola. Study of the Indus script. In *Proceedings of the International Conference of Eastern Studies*, volume 50, pages 28–66, 2005.

[27] A. Parpola. *Deciphering the Indus script.* Cambridge University Press, 2009.

[28] S. Parpola. *Etymological Dictionary of the Sumerian Language*, volume 1 and 2. Foundations for Finnish Assyrological Research, Helsinki, Finnland, 2016.

[29] R. P. Rao, N. Yadav, M. N. Vahia, H. Joglekar, R. Adhikari, and I. Mahadevan. Entropic evidence for linguistic structure in the Indus script. *Science*, 324(5931):1165–1165, 2009.

[30] R. P. Rao, N. Yadav, M. N. Vahia, H. Joglekar, R. Adhikari, and I. Mahadevan. A Markov model of the Indus Script. *Proceedings of the National Academy of Sciences*, 106(33):13685–13690, 2009.

[31] S. R. Rao. *The Decipherment of the Indus Script.* Asia Publishing House, 1982.

[32] C. Renfrew. Archaeology, genetics and linguistic diversity. *Man*, pages 445–478, 1992.

[33] P. Z. Revesz. *Introduction to Databases: From Biological to Spatio-Temporal.* Springer, 2010.

[34] P. Z. Revesz. An algorithm for constructing hypothetical evolutionary trees using common mutations similarity matrices. In *Proc. 4th ACM International Conference on Bioinformatics and Computational Biology (ACM BCB)*, pages 731–734, 2013.

[35] P. Z. Revesz. Bioinformatics evolutionary tree algorithms reveal the history of the Cretan Script Family. *International Journal of Applied Mathematics and Informatics*, 10:67–76, 2016.

[36] P. Z. Revesz. A computer-aided translation of the Cretan Hieroglyph script. *International Journal of Signal Processing*, 1:127–133, 2016.

[37] P. Z. Revesz. A computer-aided translation of the Phaistos Disk. *International Journal of Computers*, 10:94–100, 2016.

[38] P. Z. Revesz. A mitochondrial DNA-based model of the spread of human populations. *International Journal of Biology and Biomedical Engineering*, 10:124–133, 2016.

[39] P. Z. Revesz. Establishing the West-Ugric language family with Minoan, Hattic and Hungarian by a decipherment of Linear A. *WSEAS Transactions on Information Science and Applications*, 14:306–335, 2017.

[40] P. Z. Revesz. A translation of the Arkalochori Axe and the Malia Altar Stone. *WSEAS Transactions on Information Science and Applications*, 14(1):124–133, 2017.

[41] P. Z. Revesz. Sumerian contains Dravidian and Uralic substrates associated with the Emegir and Emesal dialects. *WSEAS Transactions on Information Science and Applications*, 16(1):8–30, 2019.

[42] P. Z. Revesz, M. P. Rashid, and Y. Tuyishime. The design and implementation of AIDA: Ancient Inscription Database and Analytics system. In *Proceedings of the 23rd International Database Engineering and Applications Symposium*, 2019.

[43] B. Wells. *An introduction to Indus writing.* University of Calgary, 1998.

[44] B. Wells and A. Fuls. Online Indus Writing Database. http://caddy.igg.tu-berlin.de/indus/welcome.htm, 2017.

[45] B. K. Wells. *Epigraphic Approaches to Indus Writing.* Oxbow Books, 2011.

[46] B. K. Wells and A. Fuls. *The Archaeology and Epigraphy of Indus Writing.* Archaeopress, 2015.

[47] W. Yang, L. Jin, and M. Liu. Chinese character-level writer identification using path signature feature, DropStroke and deep CNN. In *Proceedings of the 13th International Conference on Document Analysis and Recognition*, pages 546–550. IEEE, 2015.

[48] M. L. Yann and Y. Tang. Learning deep convolutional neural networks for X-ray protein crystallization image analysis. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1373–1379. AAAI Press, 2016.

[49] A. R. Zide and K. V. Zvelebil. *The Soviet Decipherment of the Indus Valley Script: Translation and Critique*, volume 156. Walter de Gruyter, 1976.

A fast solution for bi-objective traffic minimization in geo-distributed data flows

Anna-Valentini Michailidou
Aristotle University of Thessaloniki
Greece
annavalen@csd.auth.gr

Anastasios Gounaris
Aristotle University of Thessaloniki
Greece
gounaria@csd.auth.gr

ABSTRACT

Geo-distributed analytics is becoming an increasingly commonplace as IoT, fog computing and big data processing platforms are nowadays integrating with each other. In this work, we deal with a problem encountered when complex Spark workflows run on top of geographically dispersed nodes, either data centers or individual machines. There have been proposals that optimize the execution of such workflows in terms of the aggregate traffic generated or the latency (which is due to data transmission), or both metrics. However, the state-of-the-art solutions that target both objectives are either significantly sub-optimal or suffer from high optimization overhead. In this work, we address this limitation. The main solutions that we propose are both efficient and effective; based on either the extremal optimization or the greedy algorithm design paradigm, they can yield significant improvements having an optimization overhead of a few tens of seconds even for Spark workflows of 15 stages running on 15 distributed nodes. We also show the inadequacy of evolutionary optimization solutions, such as genetic algorithms, for our problem.

CCS CONCEPTS

• **Information systems** → **Query optimization**; • **Computer systems organization** → **Distributed architectures**;

KEYWORDS

data flows, workflows, Spark, optimization

ACM Reference Format:
Anna-Valentini Michailidou and Anastasios Gounaris. 2019. A fast solution for bi-objective traffic minimization in geo-distributed data flows. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331076.3331107

1 INTRODUCTION

Geo-distributed analytics, such as fog computing solutions [1, 22], is an emerging area boosted by the maturity of big data analytics platforms supporting data streams, e.g., Flink [8] and Spark [2], along with the prevalence of IoT devices in modern applications [5, 7, 18]. The execution model builds upon and extends the one in distributed [21] and parallel [9] databases. In short, the execution plan is typically a directed acyclic graph of operators and benefits from the main types of query plan parallelism, namely partitioned, pipelined and independent.

In this work, we consider Spark running over separate physical nodes with distinct data transmission capacities; as reported in [10], the applications of such a setting span several fields, such as climate science, multinational companies, bio-informatics and log analysis. Spark execution plan inherently benefits from pipelined and partitioned parallelism [3] with the underlying cluster management layers, e.g., YARN, Mesos and so on, being responsible for the actual runtime task scheduling. A typical assumption is that the cluster on which the execution runs is characterized by abundant memory and fast node interconnection speeds, and the whole processing takes place in a single geographical area. However, this assumption becomes a limitation, when the data to be processed are physically stored in multiple places and/or processing needs to occur close to the data source. To overcome this limitation, several geo-distribution-aware extensions to MapReduce-based solutions have been proposed [10].

Optimization techniques for geo-distributed Spark execution plans directly affect the manner partitioned parallelism is enforced through specifying the portion of the tasks in each Spark stage that each processing node should become responsible for. Current techniques to this end aim to minimize either the total traffic between the nodes, e.g.,[27], or the latency, e.g. [23]. In a recent previous proposal of ours, we present bi-objective solutions that target both criteria [19]. The proposal in [19] challenges the validity of a main motivation behind geo-distributed data flows, namely that it is too costly to gather data in a single place, e.g., [10, 16, 28], and is tailored to multi-stage workflows rather than simple two-stage MapReduce ones. It comprises two techniques: a greedy one that is fast but not very effective in terms of the quality of the derived solutions and another one based on iterated local search that is effective but takes longer time, in the order of couple of minutes, to compute the proposed task distribution.

In this work, we make a twofold contribution. Firstly, we combine the best attributes of the solutions mentioned above. The main bi-objective solutions that we propose are capable of running much faster that the best performing one in [19] and still yield significant improvements over the main competitor, as evidenced by the results of a thorough evaluation. The solution is based on either the extremal optimization (EO) paradigm [4, 6] or the greedy algorithm design strategy but in a less shortsighted than in [19]. Secondly, also in light of the well-known *No Free Lunch* theorem [29], it is important to find which optimization paradigm fits better to

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331107

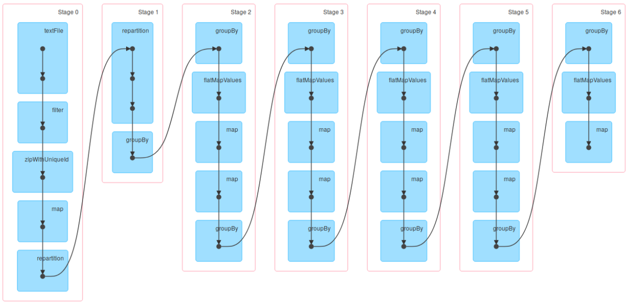


Figure 1: A real Spark DAG

our specific problem; to this end, we show that evolutionary optimization solutions, such as genetic algorithms, are inferior to the solutions we propose hereby.

Paper structure. The remainder of the paper is structured as follows. In the next section, we give a motivation scenario. In Section 3, we give the formal problem definition and we outline the solutions in [19] to make this work self-contained. We present our new solution in Section 4. The evaluation aims to cover a wide range of scenarios and is presented in Section 5. We conclude with the discussion of the related work and the open issues in Sections 6 and 7, respectively.

2 A MOTIVATION EXAMPLE

Our work is highly inspired by performance issues in modern data analytics platforms, such as Spark, which is arguably the most-widespread framework for data-intensive cluster computing to date. The distinctive feature of our work is that we do not assume a centralized, homogeneous setting; on the contrary we consider that a cluster may consist of physical machines that are geo-distributed, have heterogeneous uplink and downlink speed capacities, and communicate through sending data across a network in order to complete an application. Our algorithms can make such applications run faster by offering a task placement plan that minimizes the data transfer over the network, while we consider both of these two objectives. Next, we showcase how our algorithms, namely Greedy-full and Extremal to be presented in Section 4, can improve a Spark application.

In a geo-distributed setting, it is reasonable to assume that data transmission is the dominant factor for the application latency. Focusing on the data transmission capacities, we employ three machines (noted as M1, M2 and M3 in Table 1) with uplink speeds of 5 MB/sec, 2 MB/sec, and 5 MB/sec, respectively. The downlink speeds are 5, 3 and 2 MB/sec, respectively. The execution plan of the application we try to optimize, in the form of a Directed Acyclic Graph (DAG), is a linear one, as shown in Figure 1. Each node (bounded rectangle in the figure) is a stage that consists of tasks, the placement of which is decided by our algorithms. The edges between the nodes represent the data movement between the stages. The overall input is set to 287.6 MB and the selectivity between the stages is always equal to 1; i.e., the total amount of data being reshuffled and flowing across the stages remains the same.

We first compute the task allocation offline and then enforce the task allocation in Spark. Then, we compare the estimated running time reduction with the actual one. The offline computation ignores the CPU overhead that a real setting has even when transmitting data [20] and thus the time it refers to is only the overhead of moving data. Note that the data movement reduction is the same in the offline computation and the real run. Table 1 shows the allocations decided by each algorithm, namely Iridium [23], our main competitor, *Extremal* and *Greedy-full* (i.e., the contributions of this work) for each stage and for each machine. Note that for the first two stages of Figure 1, we do not choose a new placement because we assume that the initial data placement, on which the task placement of the first two stages depends, is fixed; these two stages just read and parallelize the initial dataset evenly. Thus we start from the task placement of Stage 2. With these new task allocations, Extremal is estimated to achieve a 50.5% reduction in the running time over Iridium, while Greedy-full achieves 9.86% reduction. The real reduction achieved in the Spark environment was 47.75% and 13% respectively, indicating that our algorithms can indeed reduce the running time of a real application in Spark. More importantly, the amount of data transmitted over the network drops by 74.97% due to Extremal (from 799.7 to 200.1MBs) and by 25.3% due to Greedy-full (from 799.7 to 597.3MBs).

Implementation Details. In order to enforce our task placement in the Spark engine, we have rebuilt Apache Spark 2.3.2 with the following changes; we override the `TaskSchedulerImpl` class, where we disable the shuffling of the offers the executors make for a task and we edit the `TaskSetManager` class to set the task locality to ``Any'' and thus prevent Spark from deciding a placement for the tasks based on the data location. Finally, we can easily emulate a geo-distributed setting with machines characterized by different downlink and uplink speeds by using machines connected to a local network and set the bandwidth limits of the executors using a tool, such as the *Wonder Shaper script*[1]

3 BACKGROUND

We first present the problem statement, which is kept the same as in [19], and then we present the two existing solutions, the strong points of which we combine in this work. The problem is stated in a system-agnostic manner; i.e., it is not applicable to Spark solely.

3.1 Problem Statement

A geo-distributed data flow is represented as a DAG $G(V, E)$. Each node $v_j \in V$, where $j = 1 \dots N$ and $N = |V|$, represents a *job* and each edge represents a shuffle data movement between the jobs. For example, in Spark data flows, we consider a job to be a *Spark stage* (note that Spark uses the terminology job to refer to a set of stages); in between such stages, data shuffling takes place. Each job runs in parallel in M data centers (DCs): i.e., each DC becomes responsible for a fraction of the job execution with the magnitude of the fraction devised by our algorithms. DCs generalize the notion of physical machines used in the motivation example.

Conceptually, the workload of a job is split into small units of work, each allocated to a specific processing element, e.g., a multi-core server of a specific DC, as an atomic unit. We refer to these

[1] available from https://github.com/magnific0/wondershaper

Table 1: Task placement decision (proportion of tasks allocated) of Iridium, Extremal and Greedy-full for each stage and machine

Algorithm/Stage-Machine	stage 2			stage 3			stage 4			stage 5			stage 6		
	M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3
Iridium	0.286	0.429	0.285	0.188	0.529	0.283	0.098	0.628	0.274	0.038	0.717	0.245	0.208	0.792	0.0
Extremal	0.0041	0.993	0.0029	0.0	0.995	0.005	0.003	0.997	0.0	0.002	0.998	0.0	0.001	0.999	0.0
Greedy-full	0.333	0.477	0.19	0.116	0.606	0.278	0.032	0.706	0.262	0.0	1.0	0.0	0.0	1.0	0.0

splits as *tasks*. Due to shuffling, in the generic case, it is necessary to move data between DCs before the execution of each task. This data movement is the dominant factor regarding the running time of the jobs, while the actual execution time of the job is considered to be negligible.

In this work, we deal with the allocation of sets of tasks to each DC for each job. Let I^j be the input dataset size of v_j. If the selectivity of the job is α^j, then the output dataset is of size $S^j = \alpha^j * I^j$; the job selectivity is defined as the ratio of its output to input size. If v_j has outgoing edges in G, S^j is divided into M parts to be sent to the jobs downstream, denoted by $r_i^j S^j,\ i = 1 \ldots M$, s.t. $\sum r_i^j = 1$. Essentially, r_i^j corresponds to the fraction of tasks of the *children nodes* of v_j assigned to the i^{th} DC (tasks are assumed to be infinitesimally divisible). In other words, r_i^j values affect the workload allocation of jobs v_k, where $(j, k) \in E$. Overall, each DC has to transfer a fraction of $(1 - r_i^j)$ of its local output data S_i^j, and to receive a total of $r_i^j * (S^j - S_i^j)$ data from all the other DCs.[2] Following the rationale in [23], we specify the uplink (resp. downlink) bandwidth of the i^{th} DC as U_i (resp. D_i). Table 2 summarizes the main notation.

Based on the above, the time for a site to send data regarding the output of a job is $TU_i^j = (1 - r_i^j) * S_i^j / U_i$, and the time to receive data is $TD_i^j = r_i^j * (S^j - S_i^j)/D_i$. The running time RT_j of v_j is $max\{TU_i^j, TD_i^j\}$.

The total data movement from a node v_j is equal to $DM_j = \sum_{i=1}^{M}(1 - r_i^j) * S_i^j$. The total data movement is $DM(G) = \sum_{j=1}^{N} DM_j$, where v_j has at least one outgoing edge.

The running time of a G, $RT(G)$ is the maximum sum of RT_j values across any path from a source job (v_j without incoming edges) to a sink one (v_j without outgoing edges); sink nodes have zero running time by default.

More formally, the problem we target is defined as follows:

Problem Statement: Given a dataflow G, a fixed distribution of the initial data across M DCs, and a running time value $RTbase$, compute the r_i^j values s.t. $DM(G)$ is minimized and $RT(G)$ is always less than $(1 + \varepsilon)RTbase$, where ε is a small constant $\varepsilon > -1$. If $0 > \varepsilon > -1$, then we enforce the solutions to seek improvements regarding both $DM(G)$ and $RT(G)$; when ε is positive, we tolerate increases in $RT(G)$ compared to $RTbase$. We can also regard positive values of ε as the percentage of the performance degradation that is tolerated.

Table 2: Notations used in the paper.

Symbol	Meaning
$G(V, E)$	the data flow DAG
N, M	number of jobs and DCs
I^j	amount of input data of a job $v_j \in V$
α^j	selectivity of a job v_j
S^j	amount of intermediate output data of a job ($S^j = \alpha^j * I^j$)
U_i	uplink bandwidth on DC i
D_i	downlink bandwidth on DC i
S_i^j	amount of intermediate data of v_j on DC i
r_i^j	fraction of tasks executed on DC i for jobs succeeding v_j
TU_i^j, TD_i^j	running time of intermediate data transfer on up and down link of DC i
$RT(G)$	total running time of G
$DM(G)$	total data movement between DCs in G
RT_j	running time of job v_j
DM_j	total data movement between DCs of job v_j
$allocations$	A $N \times M$ array holding in each row $allocations[j]$ the $r_i^j,\ j = 1 \ldots N,\ i = 1 \ldots M$ values

Note that the higher we set ε, the more the problem tends to be a single-objective optimization (that of minimizing $DM(G)$) in practice.

3.2 Existing Solutions and Limitations

In [19], a two-step approach was followed:

1. Use Iridium [23] as the guideline for the initial assignment of tasks, i.e., computation of the r_i^j values, to the DCs. Iridium decides the allocation for each job separately, after performing a topological sorting on G, and considers the nodes from the upstream to the downstream ones. In this way, $RTbase$ is derived.
2. Re-arrange the allocations with a view to decreasing the total movement cost while not allowing running time degradation more than ε times.

Then, for the second step, two techniques were proposed. The first one, is a fast greedy one. In the next section, we introduce another greedy technique explaining the differences. The second one is an Iterated Local Search (ILS) algorithm that uses Stochastic Hill Climbing (SHC) internally. It randomly perturbs the initial solution, and then looks for additional randomly chosen small changes in the perturbed configuration, so that $DM(G)$ improves,

[2] Note that in general, $S_i^j \neq r_i^j S^j$, i.e., the distribution of the intermediate results in a job is not necessarily the same as the way these results are shuffled in the next jobs. However, assuming a uniform distribution of results, it holds that $S_i^j = mean(r_i^k) * S^j$, where $(k, j) \in E$.

Algorithm 1 Greedy-full algorithm

```
Require: $allocations, RTthreshold, DM(G), RT(G), iterations$
  $bestAllocations \leftarrow allocations$
  $bestRT \leftarrow RT(G)$
  $bestDM \leftarrow DM(G)$
  for $i \leftarrow 1$ to iterations do
    for each job do
      $bottleneckDC \leftarrow$ findBottleneckDC(job)
      Reallocate tasks regarding the current job through distributing a proportion of $\beta$ of bottleneckDC's fraction to the other DCs
      $tempAllocations \leftarrow$ apply changes to all downstream jobs in $G$
      Calculate $RT(G)'$ using $tempAllocations$
      Calculate $DM(G)'$ using $tempAllocations$
      if $DM(G)' < bestDM$ && $RT(G)' \leq RTthreshold$ then
        $bestAllocations \leftarrow tempAllocations$
        $bestRT \leftarrow RT(G)'$
        $bestDM \leftarrow DM(G)'$
      end if
    end for
  end for
  return $bestAllocations, bestRT, bestDM$
```

while $RT(G)$ remains under the threshold. The ILS-based solution is shown to be capable of yielding much better results at the expense of overhead that is higher by an order of magnitude; e.g., in large flows it took 2-3 minutes on a modern PC to check 75 random perturbations, each running SHC 75 times. The extremal optimization-inspired technique and the new greedy that we introduce in the next section manage to achieve similar quality in the results running much closer to the initial greedy technique, as discussed in the experiments.

4 OUR PROPOSAL

The aim is to devise fast algorithms being as effective as the ILS-one in [19].

4.1 A greedy solution that is less shortsighted

The first algorithm we implemented is a greedy one described in Algorithm 1 (termed *Greedy-full*). The algorithm works using an initial solution derived by Iridium [23] (we also examine using a random solution in Section 5.2.3). The input of the algorithm is (i) the initial allocation of tasks on the DCs, (ii) the running time threshold $RTthreshold = (1 + \varepsilon)RTbase$, where $RTbase$ is the initial $RT(G)$, (iii) the initial $DM(G)$, (iv) the initial RT(G) and (v) the number of iterations. The output is the new allocation of tasks optimized for lower $DM(G)$ with the new $RT(G)$ to be under the threshold.

The algorithm consists of two loops. The external one is repeated 20 times while the internal one iterates over all the jobs. The number of the external iterations is configurable but unless otherwise stated, we set them to 20 (see Section 5.2.2). For each job in topological order it finds the bottleneck DC. More specifically, the *findBottleneckDC(job)* function in the algorithm returns the DC that has the least, non zero, task placement ratio. For this task placement ratio, the algorithm further removes a proportion of β and distributes it to the rest of the DCs that already have tasks proportionally. In this work, we set β equal to 1/3. Then, it assesses the global impact of such a local change. It re-calculates the task placement of the downstream nodes and if the new $RT(G)$ is under the threshold and the $DM(G)$ is minimized, then the solution becomes the best one.

In our previous work [19], we also implemented a greedy algorithm. The main difference with the algorithm of this work is that, when a job's task placement is altered, the affects are not transferred to the downstream nodes in the internal loop; i.e., the initial greedy solution focuses on local changes in a shortsighted manner. However, addressing this limitation comes at the expense of higher optimization times to derive the final task allocation, but, as shown later, the trade-off is interesting.

4.2 An EO-based solution

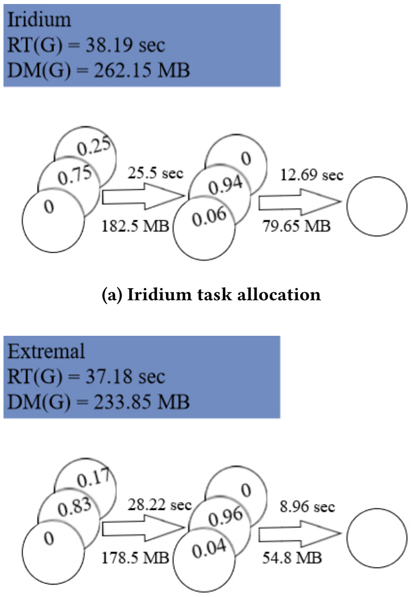
We propose an EO-based solution that will be referred to as *Extremal* (see Algorithm 2). Extremal uses also an initial solution, like Greedy-full. The input and the output remain the same for the two algorithms. Algorithm 2 consists of one loop. In each iteration, it finds the slowest job of the graph (through the *findSlowestJob(G)*) and rearranges its task placement fractions by removing a β fraction of the task ratio of randomly picked DCs. We set β equal to 1/3 and the probability is set to 1/2. This reallocation affects the downstream nodes as the S^j is re-arranged to the DCs; this reallocation is computed using the Linear Programming technique (LP) from [23]. Then the new $RT(G)$ and $DM(G)$ are calculated and the solution becomes the best one so far only if the $DM(G)$ improves and if the $RT(G)$ is under the given threshold. The number of the iterations is configurable but unless otherwise stated, we set them to 100 (see Section 5.2.2). Compared to the Greedy-full solution, its main difference is that it focuses on the slowest job overall rather than examining all jobs in turn; then, for the slowest jobs, examines more extensive random changes.

4.2.1 Example. Suppose a linear G with three nodes and three DCs on which each node is executed in parallel. The uplink and downlink of the DCs are U=(10, 1, 10), D=(10, 5, 5). The S_i^1 values are S_i^1=(120, 100, 50) and α =1 for both jobs. Figure 2a shows the result of Iridium. In the figure, each circle corresponds to a job-DC pair annotated by the corresponding r_i^j value.

We execute the loop of Algorithm 2 a single time. We set ε = 0.1. Thus the threshold is set to $RTthreshold = (1 + 0.1)38.19 =$ 42 sec. First, the algorithm searches for the slowest job which in that case, is the first one. Then, it chooses a random DC that has a fraction of tasks larger than 0, let's say that this DC is the third one. Then the algorithm removes 1/3 of its workload and transfers it to the 2nd DC, which is the only other DC with non-zero allocation; this results in r_i^1=(0, 0.83, 0.17), which, in turn yields S_i^2=(0, 224.1, 45.9) and r_i^2=(0.04, 0.96, 0). The new $RT(G)$ is 37.18 sec. The benefit in the data movement is 28.3 MBs (Figure 2b). This new $RT(G)$ is under the threshold so the solution is accepted. The final reduction over Iridium is 2.6% in terms of $RT(G)$ and 11% in $DM(G)$ which is the main metric we try to minimize. In this example, Greedy-full can reach the same outcome as Extremal, but the latter has

Algorithm 2 Extremal algorithm

```
Require: $allocations, RTthreshold, DM(G), RT(G), iterations$
  $bestAllocations \leftarrow allocations$
  $bestRT \leftarrow RT(G)$
  $bestDM \leftarrow DM(G)$
  for $i \leftarrow 1$ to iterations do
    $slowestJob \leftarrow$ findSlowestJob(G)
    for $eachDC$ do
      With probability $p$, reallocate tasks regarding the slowest
      job through distributing a proportion of $\beta$ of DC's fraction
      to the other DCs
    end for
    $tempAllocations \leftarrow$ apply changes to $G$
    Calculate $RT(G)'$ using $tempAllocations$
    Calculate $DM(G)'$ using $tempAllocations$
    if $DM(G)' < bestDM$ && $RT(G)' \leq RTthreshold$ then
      $bestAllocations \leftarrow tempAllocations$
      $bestRT \leftarrow RT(G)'$
      $bestDM \leftarrow DM(G)'$
    end if
  end for
  return $bestAllocations, bestRT, bestDM$
```



(a) Iridium task allocation

(b) Extremal algorithm task allocation

Figure 2: Example using extremal algorithm

performed only one reallocation (for the first job) while Greedy-full has checked all the jobs.

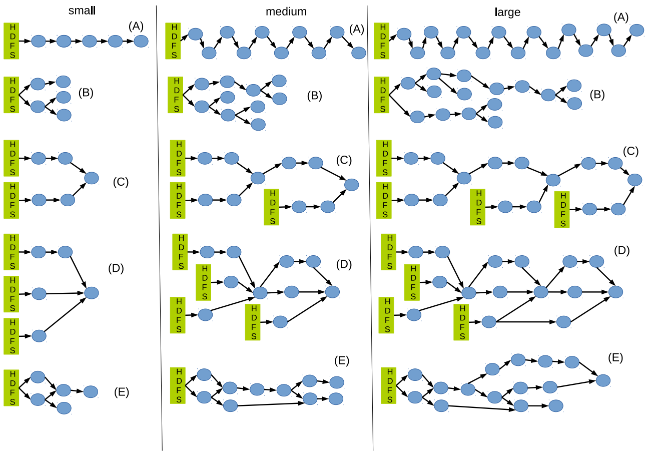


Figure 3: DAGs considered in the experiments (taken from [11])

5 EXPERIMENTS

5.1 Setting

We have already shown in Section 2 that estimated improvements correspond to improvements in real runs as well. To cover a broad range of scenarios, we resort to simulations. We use the simulation setting presented in our previous work [19], which includes five types of DAGs from [11] (presented in Figure 3) in three sizes each. The DAGs cover a very broad range of real applications, including DAGs produced when running TPC-H on Spark. To allow for a direct comparison against the results in [19], we experiment with 3 values of $M = 5; 10; 15$ and 3 values of $\varepsilon = 0.1$ and 0.2 and 0.5. The experiments were performed for every combination of DAG, number of DCs and ε value. Unless otherwise stated, $p = 0.5$, $iterations = 20$ for Greedy-full and 100 for Extremal, and $\beta = 1/3$. For the remainder of the variables, we resort to a setting similar to the one in [23]. The initial dataset I^j of the source nodes is randomly generated in the range [100MB, 1GB]. The U_i and D_i of each DC fall into the range of [100MB/sec, 2GB/sec]. The selectivities α of the jobs are between 0.01 and 2 with 50% of the job selectivities ranging from 0.01 to 0.5, 25% of them ranging from 0.5 to 1 and the rest 25% ranging from 1 to 2 (similar to the selectivities in Facebook production analytics according to [23]). For each combination of DAG type, M and ε, we created random instances according to the parameters above, and we report the average values.

5.2 Main Experiments

5.2.1 Main comparison. In the first set of experiments, we compare our algorithms namely Extremal and Greedy-full to the ones presented in our previous work [19], Iterated Local Search and Greedy, regarding the reduction in $RT(G)$ and $DM(G)$ they achieve over Iridium, when we set $\epsilon = 0.2$. The results are presented in Figure 4 and Figure 5 for $DM(G)$ and $RT(G)$, respectively. On average, Extremal reduces Iridium's $DM(G)$ by 28.16%, Greedy-full by

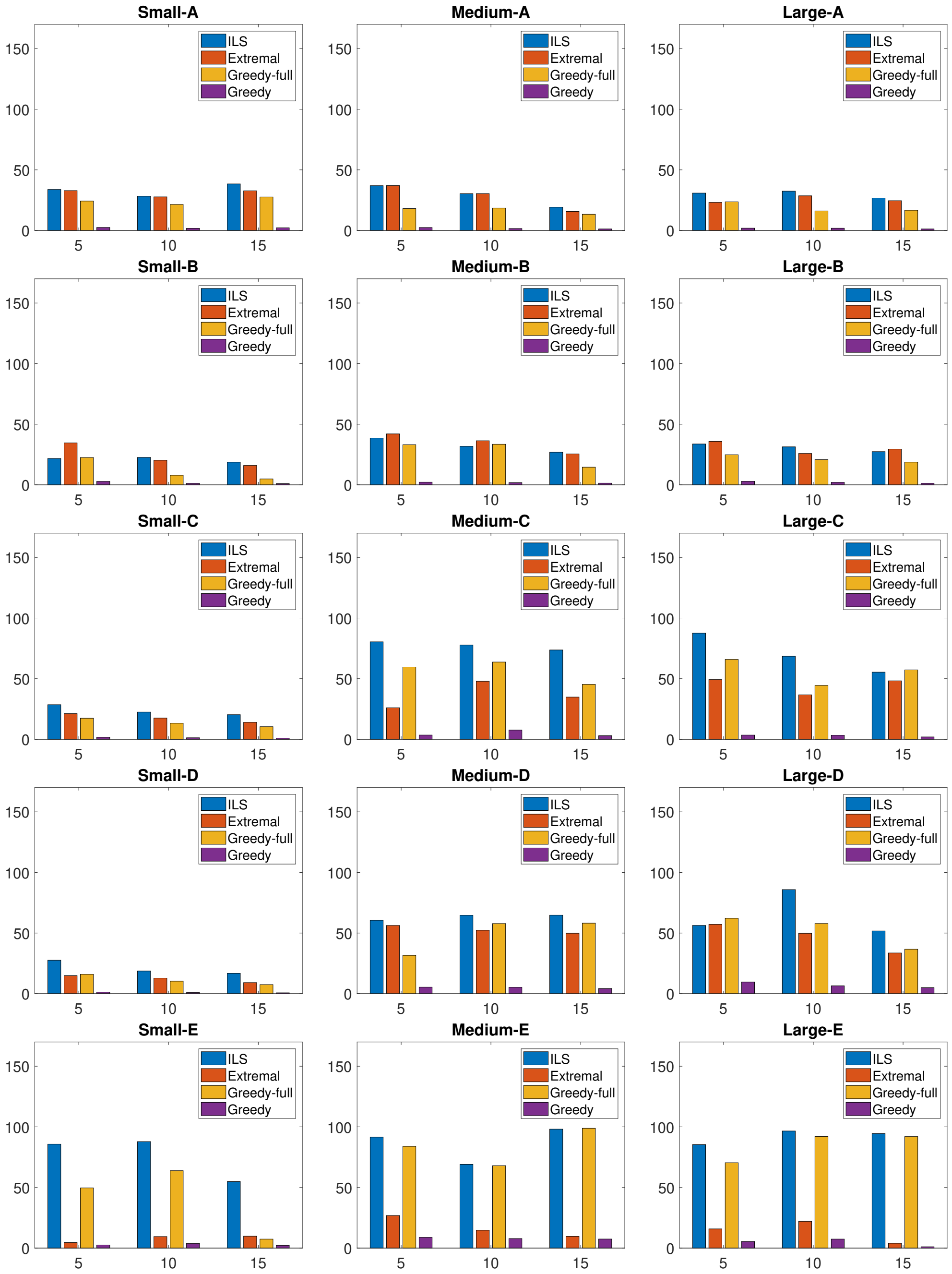


Figure 4: Percentage of $DM(G)$ reduction for M =5, 10 and 15 when ε=0.2.

37.83%, ILS by 50.12% and Greedy by 3.25%. In most cases, Greedy-full reduces the $RT(G)$ as well by a mean reduction of 28.26%, Extremal by 11.03%, ILS by 44.31%, while Greedy increases the $RT(G)$ by 7.5%.

As we can observe from Figure 4, Extremal is outperformed by ILS and Greedy-full by a large margin in the DAGs where the slowest job turns out to be one close to the sink nodes, e.g., Small E, where a single node collects data from two previous nodes. In the other cases, the behavior of Extremal and ILS is similar, whereas there exist several combinations of DAG types and sizes where Extremal is the best performing approach in average. The performance of Greedy-full is closer to that of ILS, but, as explained later, with slightly higher overhead than Extremal.

5.2.2 *Convergence Rate.* In this section, we compare the convergence rate of Extremal and Greedy-full. In order to find the

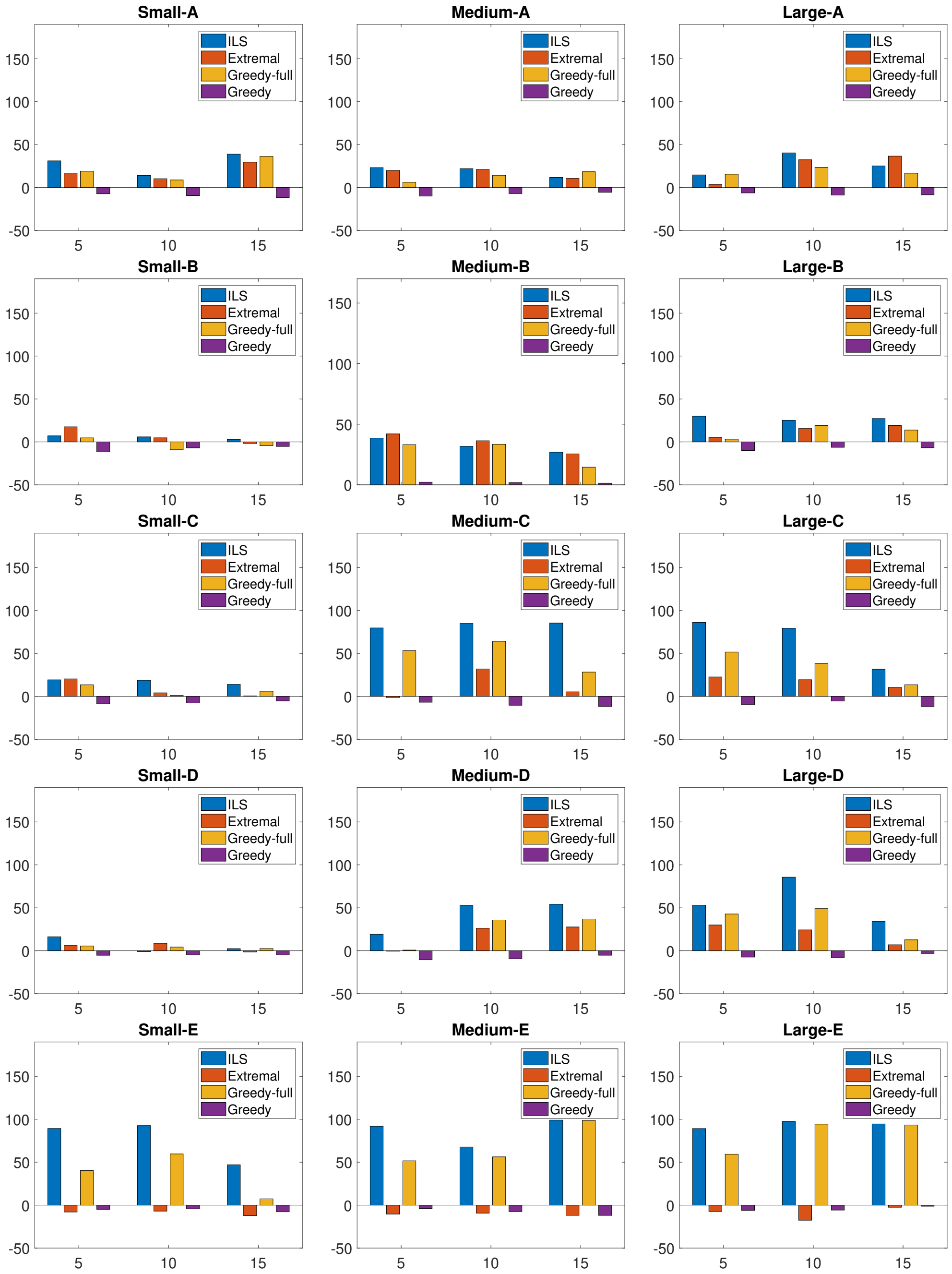


Figure 5: Percentage of $RT(G)$ reduction for M =5, 10 and 15 when ε=0.2.

convergence rate of Greedy-full we set the number of iterations to $N * M = 6 * 10 = 60$ while Extremal iterates 100 times. Figure 6 shows the results for the Small-A DAG and 10 machines. We can observe that Greedy-full converges at around the 10th iteration, which is faster than Extremal which converges at around 50th iteration. We should also consider the running time of the algorithms. While Extremal iterates more times, it only takes 5.7 sec but Greedy-full takes 13.08 sec. Taken that into consideration, Extremal converges at around 2.85 sec and Greedy-full at around 2.18 sec (machine specifications are given when discussing time overheads in more detail). This explains our choice to set the number of iterations of the two algorithms to 20 and 100, respectively. We further investigate the behavior of Extremal, ranging the number of iterations from 25 to 150. The results are presented in Figures 7

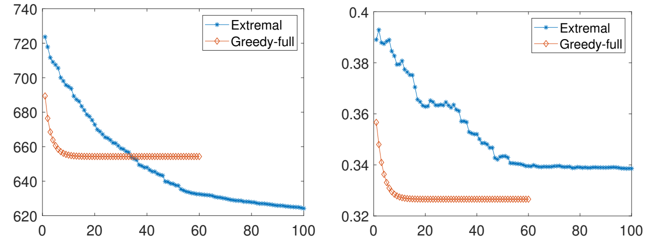


Figure 6: $DM(G)$ (left) and $RT(G)$ (right) convergence rate for the Small-A (top) DAG when running Extremal and Greedy-full (M=10, ε=10%)

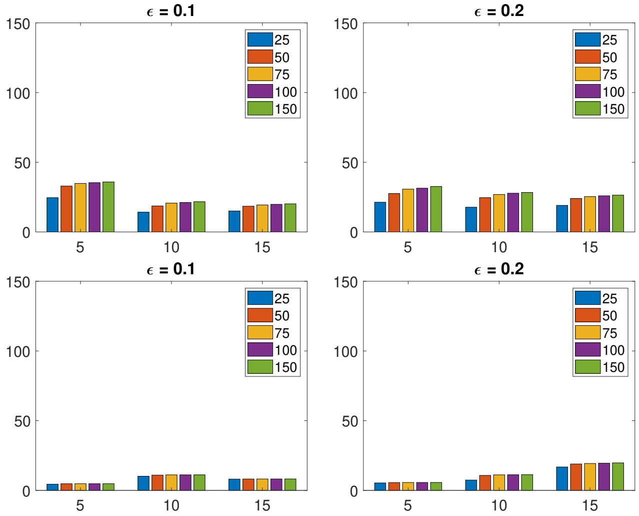


Figure 7: Percentage of $DM(G)$ reduction for the Small-A (top) and Large-E (bottom) DAGs when running Extremal for different M (horizontal axis), ε and number of iterations

and 8. Setting the iterations to 100 offers a good trade-off between the quality of the output and the running time of the algorithm.

5.2.3 Impact of initial allocation. In this set of experiments, we tried initializing the Greedy-full and Extremal algorithms with a random solution rather than the Iridium one. The results show that the algorithms are quite sensitive to the initial allocation as they cannot produce a plan that improves on the Iridium's $RT(G)$ and $DM(G)$ (no figures are shown due to space constraints). In most cases, the final results of the algorithms that were initialized with the random solution are worse than the Iridium ones. Therefore, the initialization phase in our solution that first optimizes for RT (though employing Iridium's approach) and then proceeds to DM minimization is crucial in the bi-objective optimization solution.

5.2.4 Time overheads. The running time of each algorithm is presented in Table 3. The experiments were performed on a machine with i7-4510U CPU at 2.00GHz with 8 GB of RAM. Two main observations can be drawn: (i) Extremal and Greedy-full incur lower overhead than ILS by an order of magnitude; and (ii) Greedy-full is slower than Extremal regarding non-small flows.

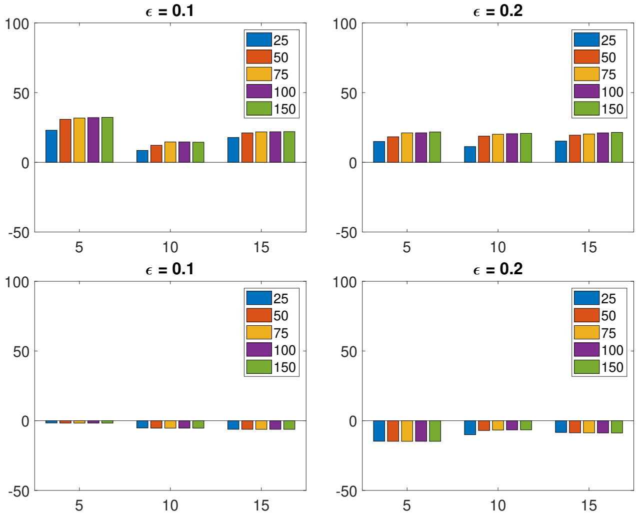


Figure 8: Percentage of $RT(G)$ reduction for the Small-A (top) and Large-E (bottom) DAGs when running Extremal for different M (horizontal axis), ε and number of iterations

Algorithm 3 Genetic algorithm

```
Require: $populationSize, recombinationProb, mutationProb, generations$

  $population \leftarrow initializePopulation(populationSize)$
  $best, bestRT, bestDM \leftarrow getBest(population)$
  for $i \leftarrow 1$ to $generations$ do
    $parents \leftarrow$ selectParents(population)
    $children \leftarrow \varnothing$
    for each pair in parents do
      $child1, child2 \leftarrow recombination(pair, recombinationProb)$
      $children \leftarrow mutate(child1, mutationProb)$
      $children \leftarrow mutate(child2, mutationProb)$
    end for
    $population \leftarrow combine(population, children)$
    $population \leftarrow population(1 : populationSize)$
    $best, bestRT, bestDM \leftarrow getBest(evaluatedPopulation)$
  end for
  return $best, bestRT, bestDM$
```

5.3 Comparison against an Evolutionary Solution

In this section, we present how an evolutionary algorithm performs in our setting. Specifically, we have implemented the genetic algorithm described in Algorithm 3 and compared it to Extremal and Greedy-full. First, the Genetic algorithm initializes the population and finds the best solution among them. In our implementation, we initialized the population of size 400 with the Iridium's solution, about 10% greedy solutions over Iridium and 90% random ones. Then, the population is divided into pairs from the recombination of which new solutions (children) are produced. Then, the children are mutated with a small probability, inserted in the population and the best solution is found. This is repeated for a number

Table 3: Running times of the algorithms for different M values (in sec).

	Small A			Medium C			Large E		
Algorithm \M	5	10	15	5	10	15	5	10	15
Iridium	0.13	0.17	0.18	0.28	0.29	0.3	0.3	0.31	0.36
Extremal	5.44	5.7	5.71	4.37	4.73	5.95	13.91	18.17	18.77
Greedy-full	3.66	3.99	4.12	10.8	11.39	11.81	20.93	22.32	24.51
Greedy	0.16	0.19	0.21	0.8	1.12	1.31	2.45	3.29	3.6
ILS (75 iterations)	59.99	69.7	71.98	87.73	91.16	93.44	130.98	131.26	159.04

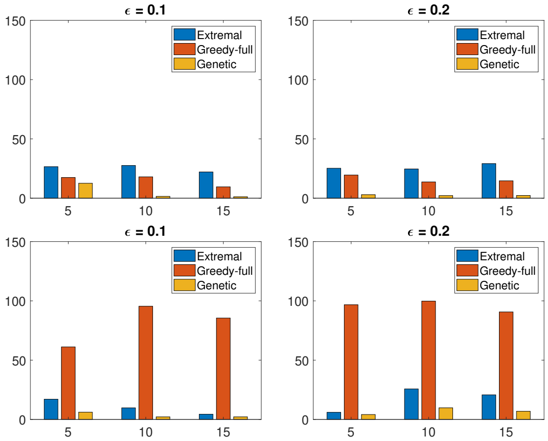


Figure 9: Percentage of $DM(G)$ reduction for the Small-A (top) and Large-E (bottom) DAGs when running Extremal, Greedy-full and Genetic for different M (horizontal axis) and ε

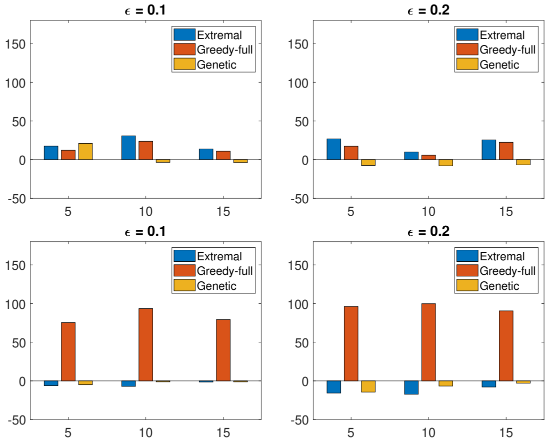


Figure 10: Percentage of $RT(G)$ reduction for the Small-A (top) and Large-E (bottom) DAGs when running Extremal, Greedy-full and Genetic for different M (horizontal axis) and ε

of iterations called generations. The output of the algorithm is the best $RT(G)$ and $DM(G)$.

Figures 9 and 10 show the results, when the number of generations is set to 400. As can be seen, Genetic is the least beneficial algorithm for our setting. That indicates that it is more effective to work on a single solution rather than having a collection of them, e.g the population in the Genetic. Moreover the random combination of components from different solutions does not lead to a good outcome either.

6 RELATED WORK

As the amount of jobs that need to be executed in geo-distributed data centers is increasing, there have been several proposals for optimized task placement. Many works focus on minimizing the total traffic. For example, WANalytics [27] deals with the task placement in this regard, but does not consider the overall running time. [17] offers a prediction of job execution time but focuses only on the minimization of the data movement as well. Clarinet [26] is a query optimizer that chooses the best execution plan among the ones provided by multiple query optimizers, considering the WAN-consumption during scheduling and task placement.

On the other hand, there are solutions that employ the minimization of the running time as their objective. Two earlier proposals, include Nebula [24] and Tetris [12] that overlook issues regarding total data movement. Heintz et al. [13] developed a framework that optimizes the data and task placement of each phase of a mapreduce job focusing on minimizing the makespan of the query but not on the overall data movement either. Iridium [23], which is the work against which we compare our solution, also focused only on the running time. However, Iridium can modify the placement of the initial data as well. In our work, we assume that initial data allocation is fixed. Tetrium [15] also tries to improve upon Iridium, as we do, in two ways. Firstly, through considering the time spent due to computations and not only data transmission. Secondly, through making scheduling decisions at a lower level than simple decision of the fraction of the tasks to run on each site to account for the case when the slots available are less than the allocated tasks. Both these extensions are interesting and we plan to investigate them in the future. Contrary to our proposal, it focuses mostly on response time but supports constraints on data movement (we treat the two metrics as of equal importance through first optimizing for response time and then for data movement); also, in our solutions, we manage to handle stage dependencies better through not running a stage-by-stage technique only once.

There are also works that consider both metrics. For example, Flutter [14] is a system that performs bi-objective task placement online but all tasks of the same stage are allocated to a single data center. Works on multi-objective query optimization, such as [25], suffer from the same limitation. Finally, the work in [30] targets both metrics but is tailored to a single MapReduce flow with the reducer being executed on a single DC. In summary, none of these works can be applied to a generic DAG, where each DAG vertex is distributed across several nodes.

7 DISCUSSION

In this work, we proposed a fast solution that decides the task placement in complex analytics workflows targeting the minimization of both response time and data movement. The thorough experiments show that we can yield significant improvements over our main competitor with much less overhead than the previous proposal to the same end.

In general, there are further open issues in the multi-objective problem we deal with. Taking into consideration the processing costs and capacity constraints of the participating nodes, in line with the work in [15], is a promising direction for future work. Also, investigating how the required metadata can be efficiently monitored online is an open issue. Finally, further research is required for taking into account aspects such as scheduling decisions when multiple workflows run on the same infrastructure concurrently.

ACKNOWLEDGMENT

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code:T1EDK-01944)

REFERENCES

[1] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodík, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. 2017. Real-Time Video Analytics: The Killer App for Edge Computing. *IEEE Computer* 50, 10 (2017), 58–67.

[2] Michael Armbrust, Tathagata Das, Joseph Torres, Burak Yavuz, Shixiong Zhu, Reynold Xin, Ali Ghodsi, Ion Stoica, and Matei Zaharia. 2018. Structured Streaming: A Declarative API for Real-Time Applications in Apache Spark. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018.* 601–613.

[3] Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, and Matei Zaharia. 2015. Spark SQL: Relational Data Processing in Spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015.* 1383–1394.

[4] Stefan Boettcher. 2000. Extremal Optimization: Heuristics via Coevolutionary Avalanches. *Computing in Science and Engg.* 2, 6 (Nov. 2000), 75–82.

[5] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC '12).* ACM, New York, NY, USA, 13–16. https://doi.org/10.1145/2342509.2342513

[6] Jason Brownlee. 2011. *Clever algorithms: nature-inspired programming recipes.* Jason Brownlee.

[7] C. C. Byers. 2017. Architectural Imperatives for Fog Computing: Use Cases, Requirements, and Architectural Techniques for Fog-Enabled IoT Networks. *IEEE Communications Magazine* 55, 8 (Aug 2017), 14–20. https://doi.org/10.1109/MCOM.2017.1600885

[8] Paris Carbone, Stephan Ewen, Gyula Fóra, Seif Haridi, Stefan Richter, and Kostas Tzoumas. 2017. State Management in Apache Flink®: Consistent Stateful Distributed Stream Processing. *PVLDB* 10, 12 (2017), 1718–1729.

[9] David J. DeWitt and Jim Gray. 1992. Parallel Database Systems: The Future of High Performance Database Systems. *Commun. ACM* 35, 6 (1992), 85–98.

[10] S. Dolev, P. Florissi, E. Gudes, S. Sharma, and I. Singer. 2017. A Survey on Geographically Distributed Big-Data Processing using MapReduce. *IEEE Transactions on Big Data* (2017), 1–1.

[11] Anastasios Gounaris, Georgia Kougka, Rubén Tous, Carlos Tripiana Montes, and Jordi Torres. 2017. Dynamic Configuration of Partitioning in Spark Applications. *IEEE Trans. Parallel Distrib. Syst.* 28, 7 (2017), 1891–1904.

[12] Robert Grandl, Ganesh Ananthanarayanan, Srikanth Kandula, Sriram Rao, and Aditya Akella. 2014. Multi-resource Packing for Cluster Schedulers. In *Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM '14).* ACM, New York, NY, USA, 455–466. https://doi.org/10.1145/2619239.2626334

[13] B. Heintz, A. Chandra, R. K. Sitaraman, and J. Weissman. 2016. End-to-End Optimization for Geo-Distributed MapReduce. *IEEE Transactions on Cloud Computing* 4, 3 (July 2016), 293–306. https://doi.org/10.1109/TCC.2014.2355225

[14] Z. Hu, B. Li, and J. Luo. 2016. Flutter: Scheduling tasks closer to data across geo-distributed datacenters. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications.* 1–9. https://doi.org/10.1109/INFOCOM.2016.7524469

[15] Chien-Chun Hung, Ganesh Ananthanarayanan, Leana Golubchik, Minlan Yu, and Mingyang Zhang. 2018. Wide-area Analytics with Multiple Resources. In *Proceedings of the Thirteenth EuroSys Conference (EuroSys '18).* ACM, New York, NY, USA, Article 12, 16 pages. https://doi.org/10.1145/3190508.3190528

[16] Konstantinos Kloudas, Margarida Mamede, Nuno Preguiça, and Rodrigo Rodrigues. 2015. Pixida: Optimizing Data Parallel Jobs in Wide-area Data Analytics. *Proc. VLDB Endow.* 9, 2 (Oct. 2015), 72–83.

[17] P. Li, S. Guo, T. Miyazaki, X. Liao, H. Jin, A. Y. Zomaya, and K. Wang. 2017. Traffic-Aware Geo-Distributed Big Data Analytics with Predictable Job Completion Time. *IEEE Transactions on Parallel and Distributed Systems* 28, 6 (June 2017), 1785–1796. https://doi.org/10.1109/TPDS.2016.2626285

[18] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. 2017. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal* 4, 5 (Oct 2017), 1125–1142. https://doi.org/10.1109/JIOT.2017.2683200

[19] Anna-Valentini Michailidou and Anastasios Gounaris. 2019. Bi-objective traffic optimization in geo-distributed data flows. *Big Data Research* https://doi.org/10.1016/j.bdr.2019.04.002 (2019).

[20] Kay Ousterhout, Ryan Rasti, Sylvia Ratnasamy, Scott Shenker, and Byung-Gon Chun. 2015. Making Sense of Performance in Data Analytics Frameworks. In *12th USENIX Symposium on Networked Systems Design and Implementation, NSDI 15, Oakland, CA, USA, May 4-6, 2015.* 293–307.

[21] M. Tamer Özsu and Patrick Valduriez. 2011. *Principles of Distributed Database Systems, Third Edition.* Springer.

[22] Pankesh Patel, Muhammad Intizar Ali, and Amit P. Sheth. 2017. On Using the Intelligent Edge for IoT Analytics. *IEEE Intelligent Systems* 32, 5 (2017), 64–69.

[23] Qifan Pu, Ganesh Ananthanarayanan, Peter Bodik, Srikanth Kandula, Aditya Akella, Paramvir Bahl, and Ion Stoica. 2015. Low Latency Geo-distributed Data Analytics. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15).* ACM, New York, NY, USA, 421–434. https://doi.org/10.1145/2785956.2787505

[24] M. Ryden, K. Oh, A. Chandra, and J. Weissman. 2014. Nebula: Distributed edge cloud for data-intensive computing. In *2014 International Conference on Collaboration Technologies and Systems (CTS).* 491–492.

[25] E. Tsamoura, A. Gounaris, and K. Tsichlas. 2013. Multi-objective Optimization of Data Flows in Multi-cloud Environment. In *Proceedings of the 2nd International Workshop on Data Analytics in the Cloud (DanaC'2013) (in conjunction with ACM SIGMOD/PODS'2013).* New York, NY, 6–10. http://delab.csd.auth.gr/papers/DANAC2013tgt.pdf

[26] Raajay Viswanathan, Ganesh Ananthanarayanan, and Aditya Akella. 2016. CLARINET: WAN-Aware Optimization for Analytics Queries. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16).* USENIX Association, Savannah, GA, 435–450. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/viswanathan

[27] Ashish Vulimiri, Carlo Curino, Brighten Godfrey, Konstantinos Karanasos, and George Varghese. 2015. WANalytics: Analytics for a Geo-Distributed Data-Intensive World. In *CIDR.*

[28] Ashish Vulimiri, Carlo Curino, P. Brighten Godfrey, Thomas Jungblut, Jitu Padhye, and George Varghese. 2015. Global Analytics in the Face of Bandwidth and Regulatory Constraints. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15).* USENIX Association, Oakland, CA, 323–336.

[29] D. H. Wolpert and W. G. Macready. 1997. No Free Lunch Theorems for Optimization. *Trans. Evol. Comp* 1, 1 (1997), 67–82.

[30] Wenhua Xiao, Weidong Bao, Xiaomin Zhu, and Ling Liu. 2017. Cost-Aware Big Data Processing Across Geo-Distributed Datacenters. *IEEE Trans. Parallel Distrib. Syst.* 28, 11 (2017), 3114–3127.

Open Government Data usage: a brief overview

Alfonso Quarati*
quarati@ge.imati.cnr.it
Institute for Applied Mathematics and Information Technology, National Research Council
Genova, Italy

Monica De Martino
demartino@ge.imati.cnr.it
Institute for Applied Mathematics and Information Technology, National Research Council
Genova, Italy

ABSTRACT

The increasingly massive spreading of Open Government Data (OGD) is hailed as a driving force for economic and social growth, as well as an essential factor in promoting public awareness of the work of institutional decision-makers. However, this high data availability can disorient users when deciding which sources are best suited to their needs. The awareness of this indecision worries the heads of OGD portals, who have to face the increasingly concrete risk that a large part of their information assets can remain unused. To assess the merits of these concerns, this document aims to provide a snapshot on the use of OGD portals based on usage indicators directly or programmatically obtainable. Considering an adequately representative sample of OGD portals, our analysis highlighted two aspects. A confirmation of the fact that most of the published datasets are very lightly used. The perception that information about the use of portals is rarely made available to the users.

CCS CONCEPTS

• **Information systems → Data analytics**; *Digital libraries and archives*; *RESTful web services.*

KEYWORDS

Open Government Data portals, Open Data usage, Views and Downloads metrics

ACM Reference Format:
Alfonso Quarati and Monica De Martino. 2019. Open Government Data usage: a brief overview. In *23rd International Database Engineering Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3331076.3331115

1 INTRODUCTION

e-Government data covers authoritative and valuable information on our society. Open Government Data[1] (OGD) usually refers to public records (e.g. on transport, infrastructure, education, health, environment) that can be used and redistributed by anyone either for free or at a marginal cost [4]. Access and free use of government data are seen as a goldmine of unprecedented social and economic potential [6][2]. However, although the exponential growth of OGD provides consumers with a massive amount of data, it also forces them to questioning on the value of these unfamiliar sources in meeting their information needs thus hampering their use. This leaves the data providers with the uneasy feeling that large part of their data remains untapped [14][15]. This concern transpires in the reports of some US Chief Data Officers (CDOs) reported in “Are Open Data Efforts Working?” published on Government Technology Magazine in March 2018 [17], with several civic leaders reached the conclusion: “We counted the clicks and we saw that these portals just weren't being used”. Although OGD are considered a driving force for transparency [7][11], they have limited value if they are not utilized [8].

To cope with such a situation, government agencies have to make sure, by monitoring users' behaviour, that people are able to directly or indirectly (e.g. through third-party applications) access to their datasets, so that they can be used to answer citizens' questions [20]. The adoption of metadata by OGD portals can help to facilitate user access through search and filtering capabilities [19][23]. In order to better inform potential users about the degree of adequacy of the data retrieved with respect to their needs, the metadata should also contain information on the quality and on the provenance of the datasets [1][13], in accordance with the W3C Web Best Practices Recommendation[2]. Moreover, metadata may report OGD usage indicators such the numbers of downloads and views, which drive users to the most popular datasets. These measures provide better insights on users' behaviour and may help policymakers to evaluate the impact of OGD resources [16].

This paper aims to provide an overview of the attractiveness of OGD portals on potential users. Initially we considered the institutional portals of 98 countries from which we investigated the presence of usage metadata directly visible to the users. Subsequently, our analysis focused on a set of six portals that also allow programmatic access via API to two usage indicators, i.e. the number of views and downloads, for each portal dataset.

The analysis suggests that most of the datasets directly reachable from the portals are little (a few dozen times at most) or rarely used. Because of this evaluation, we also provide some insights into the practices of publishing usage metrics statistics from portal managers, noting that even this type of metadata is rarely or only partially provided, making it less immediate for users to evaluate the reception of a dataset of their interest. This work is intended to be a first step towards the understanding of how and if there is any

*Corresponding Author

[1] http://www.oecd.org/gov/digital-government/open-government-data.htm

[2] https://www.w3.org/TR/dwbp/#provenance

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331115

relationship between the use of datasets and the inherent quality of the same or of the metadata associated with them.

The paper is structured as follows. Section 2 introduces the OGD portals object of our analysis and the methodological approach. Section 3 presents the results on the availability of usage metrics for a set of OGD portals and some insights on their usage. Section 4 discusses the implications of the study. Section 5 presents the conclusions and future works.

2 MATERIAL AND METHODS

The evaluation of the use of OGD is performed considering a large set of OGD portals all over the world of which the availability of usage metrics, e.g. the number of views and downloads for the datasets, the applications that re-use the datasets, supplied as metadata visible from the platforms or recoverable in a programmatic way, have been verified.

2.1 The selected OGD portals

We have selected 98 OGD portals among those of world countries examined and ranked by two initiatives: the Global Open Data Index (GODI)[3] for the assessment of the publication of public data opened from a civic perspective, and the OECD Open Useful and Reusable data (OURdata) Index on Open Government Data performed by the Organization for Economic Co-operation and Development (OECD)[4] for the assessment of the governments' efforts to implement open data in the three critical areas: openness, usefulness and re-usability of OGD [10]. We have considered the 94 countries ranked by GODI in 2017 according with different data categories. In addition, we have also considered four countries (i.e. Korea, Spain, Ireland, and Estonia) not included in the GODI ranking list, but ranked by OECD in 2018. For each countries, we have analyzed its OGD portal and the visibility of usage metadata to users. Data collection was conducted on 28 and 29 March 2019.

Figure 1 shows the synthesis of the results providing the representation of the usage metadata distribution of all the 98 OGD portals. It highlights a general lack of portals in providing the user with such metadata. Few portals provide usage information: only 10% and 6% of the countries provide respectively Views and Downloads metadata, and 2% provide other kinds of information (i.e. followers, reusing applications). 65% of the countries do not provide any usage indicators and 17% portals are still at a beginning phase of development, they do not publish any datasets or at least we could not find them.

Based on this analysis, Table 1 reports the list of portals we have considered in our study. They are the portals for which there is usage information immediately visible at metadata level. The only exception concerned the English portal: even if not immediately visible, these data has been obtained by downloading a specific CSV file. We have also included a single non-national portal managed by the United Nations Office for the Coordination of Humanitarian Affairs (UN-OCHA[5]), the Humanitarian Data Exchange portal (HDX) aimed at sharing data across crises, as it provides data of different countries such as those included in the 17% set (e.g. Venezuela, Barbados, Saint Kitts) whose OGD portals has not been found.

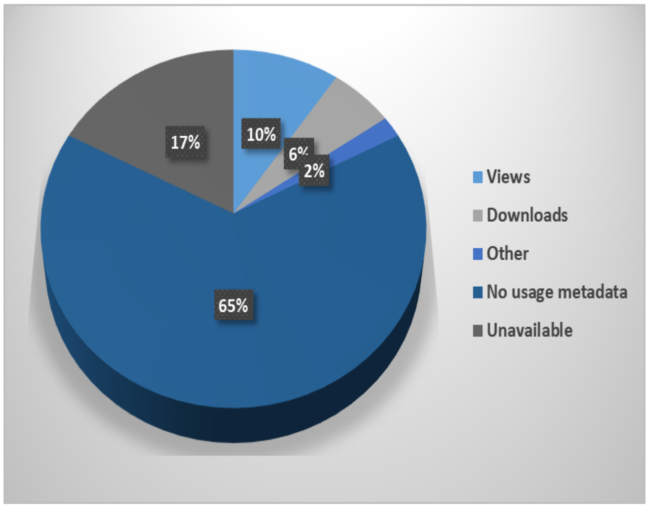


Figure 1: Usage metadata distribution of OGD portals of 98 worldwide countries

2.2 Usage Metrics

The EU commission, in the yearly published "Open Maturity in Europe" report for 2018 [3], mentions a 'Portal usage' indicator, which takes into account portal usage metrics such as the number of unique visitors, the percentage of foreign visitors, typical user profiles, traffic generated via portals API, popular data domains and the most consulted datasets. While several numbers, and related graphs, are supplied in relation to the first five metrics, just few lines are dedicated to articulate on the most *consulted datasets*: they "stem from domains that are of broad public interest, such as public spending and procurement, mobility, social economic numbers, in particular housing and environment data".

To get insights on the data demand by users we analysed two indicators: the number of online views and the number of downloads associated to every portal datasets [18]. We mean by *Views* the number of times the page of a dataset was loaded in users' browsers and by *Downloads* the number of times a user has clicked (on URL or on a 'Download' button) to retrieve a resource for a particular dataset. These values can be found in logs[6] or returned by portal APIs and can be found, along other dataset metadata, on the dataset access page.

In some way, this information accounts for the activities of *direct users*, i.e. those who access the datasets directly [15]. Perhaps, a more mature measure to assess the impact of datasets on end users could take into consideration the *indirect* users, those who use data indirectly, i.e. processed by intermediaries. These values can not be inferred from the current portals. At the most, references to applications based on the datasets contained therein are reported in specific sections of the portal, with an indication of the datasets involved, but not (at least to the best of our knowledge, with the sole

[3] https://index.okfn.org/

[4] http://www.oecd.org/gov/digital-government/open-government-data.htm

[5] https://www.unocha.org/

[6] https://www.europeandataportal.eu/sites/default/files/edp_landscaping_insight_report_n4_2018.pdf

Table 1: OGD portals and usage dimension values derived from direct portal access through Metadata (M), or downloadable file (D). Information updated at 26th March 2019

Country	Portal	#datasets	Views	Downloads	Other
U.S.	data.gov	236,352	M	-	-
UK	data.gov.uk	47,738	D	D	-
Ireland	data.gov.ie	9,001	M	-	-
France	data.gouv.fr	35,663	-	-	M
Portugal	dados.gov.pt	2,060	-	-	M
UN-OCHA	data.humdata.org	8,571	-	M	-
Taiwan	data.gov.tw	40,055	M	M	-
Colombia	datos.gov.co	10,231	M	-	-
Latvia	data.gov.lv	267	M	-	-
Poland	dane.gov.pl	1,077	M	M	-
Slovenia	podatki.gov.si	3,389	M	-	-
India	data.gov.in	265,929	M	M	-
Russia	data.gov.ru	21,878	M	M	-
Puerto Rico	data.pr.gov	179	M	-	-
Korea	data.go.kr	28,871	-	M	-

exception of the French and Portuguese portals) with the number of applications that exploit each dataset, included in the associated metadata. For this reason we have limited ourselves to recovering direct access measures.

2.3 Metric values retrieval

Retrieve the values of the usage metric can not be proceed by hand, considering every single published dataset of each portal. This information can be recovered in two ways: a) by downloading, in very specific cases, some files containing statistics about the use of each portal dataset; b) programmatically, by means of specific APIs supplied by the software Open Data platform on which the portal is built.

As regards the second option, CKAN[7] is the most widely used open source data management system [9] that provides the tools for publishing, finding and using open data. It includes also a rich RESTful JSON API for querying and retrieving datasets information. It is actually used by many governments, organizations and companies to make their huge data sources open and available. Generally, these organizations deploy their own instances of CKAN, personalizing its default user interface and providing their own data-storage to store the published datasets.

The information related to the number of views for a dataset can be obtained through CKAN API, extracting the content of a specific field called `tracking_summary`[8], which in turn contains a pair of values `total` and `recent` (i.e. Views in the last 14 days). When allowed by the portal it is possible to make a REST call that retrieves this information, returned in JSON or XML objects, through different http clients. To this end, we used the library `httr` of statistical software R. In any case, we first queried the CKAN server to retrieve the lists of the managed datasets, and only when it succeeded we sent a `GET` call to retrieve the metadata associate to each dataset. However, the presence of tracking information, in the `GET` response, is not guaranteed by default but have to be enabled server side[9]. Below the call to retrieve the tracking information related to the dataset with id='xxxxx' from data.gov.

```
ds <- GET("http://catalog.data.gov/",path="/
    api/3/action/package_show",query=list(id
    ="xxxxx"),include_tracking='T')
cds <- content(ds)
total <- cds$result$tracking_summary$total
recent <- cds$result$tracking_summary$recent
```

By cycling on the whole list of datasets of the portal the overall views situation may be recovered. Indeed, CKAN APIs only returns dataset Views and not Downloads information. A portal such as the Humanitarian Data Exchange (HDX), based on an extension to CKAN, also supply a specific R library (e.g. `rhdx`) for the recovery not only of views numbers but also downloads numbers. The following R excerpt code retrieves usage data for the third dataset of the portal:

```
ds <- search_datasets()
downloads <- ds[[3]]$data$total_res_downloads
views <- ds[[3]]$data$pageviews_last_14_days
```

Finally, some portals, like the French and Portuguese one, use other OGD platforms[10], which provide different APIs than CKAN, both in the type of call and in the type of information returned. In particular the number of views, returned in the JSON response (in a sub-field named 'views'), is not indicated in the metadata visible to users, instead two other indicators are reported namely 'Reuse number' and 'Number of followers'.

[7] https://ckan.org/

[8] From version 2.7.3 the package_show API call does not return the tracking_summary, keys in the dataset or resources by default any more

[9] https://docs.ckan.org/en/2.8/maintaining/tracking.html

[10] https://github.com/opendatateam/udata/

3 RESULTS

The analysis carried out on the portals listed in Table 1, focused on two aspects related to the use of OGD: i) the availability of usage information; ii) the portal usage trends related to the user behaviour.

3.1 Gathering usage data

For each portal listed in Table 1, we have looked for usage information obtainable via APIs. This has been only possible for the first six portals listed in the table. For the remaining nine, we have not always found the availability of this access (e.g. Poland, Slovenia, India), or when present the APIs could only be activated following a token request (e.g. Russia, Colombia). Thus, we have decided, for the moment, not to consider them.

As the first six portals accept REST calls retrieving catalogue information (i.e. answer with a 200 code and the list of all the datasets in their catalog), we proceeded at retrieving the usage information at dataset level. However, in two cases, i.e. the Portugal and UK portals, the returned usage metric values are 0 for all the datasets (and the associated resources). For this reason, we have renounced to carry out further analysis for the Portuguese portal. Instead, for the UK we have used the values contained in a CSV file[11], which provided a usage information snapshot. Unfortunately, these data are no longer available but we have used them to understand the situation of the portal in the UK.

At the end of this further skimming, the five portals from which we have been able to programmatically extract useful information are data.gov, data.gov.uk, data.gov.ie, data.gouv.fr and data.humdata.org. Although this may be considered a limited sample, we think it is however representative both in terms of the relevance of the national portals, and in terms of the number of datasets in their catalog. In fact, the statistics on the number of datasets present in the 98 portals, initially considered, are the following: median = 783, 3rd quartile = 10.173, 70%, 90% and 95% percentiles, respectively, 8.300, 40.497 and 79.149 datasets. Indeed data.gov supplies one of the largest OGD catalog, data.gov.uk and data.gouv.fr are two large size ones, and data.gov.ie and HDX fall in 30% larger. One last note: while the information obtained by API from the US, French, Irish and HDX portals provides a snapshot of the current situation, that derived from the CSV file of UK supplies a crystallized image of the datasets usage.

3.2 Portals usage trends

Views and Downloads values gathered from the five selected portals provide information about the usage frequencies for their datasets. Specifically, Figures 2, 3 and 4 show the Views frequency for the U.S., Irish and French portals, Figures 5 and 6 illustrate the Views and Downloads frequencies for the UK and HDX portals.

In all cases, the curves show heavy-tailed distributions with few datasets with a high frequency of use, and most of them with very low frequency. These results are confirmed by examining the descriptive statistics reported in Table 2, which supplies more insights on the usage trend. For what concerns the relationship between Views and Downloads, the values available for data.gov.uk confirm the expected data: the number of downloads is less than the Views; while the inverse relationship in the case of HDX should not deceive: in fact the value of the Views reported concerns the last 14 days while that of the downloads shows the overall total.

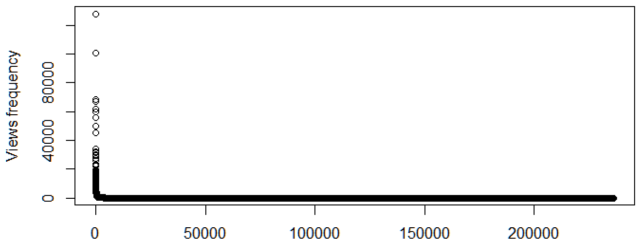


Figure 2: Views frequency for data.gov datasets

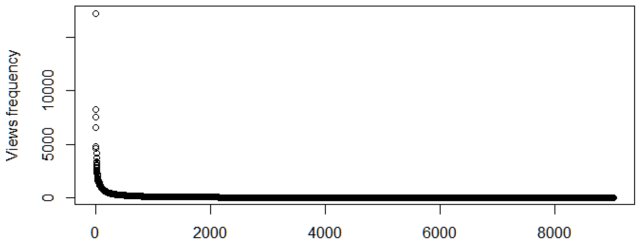


Figure 3: Views frequency for data.gov.ie datasets

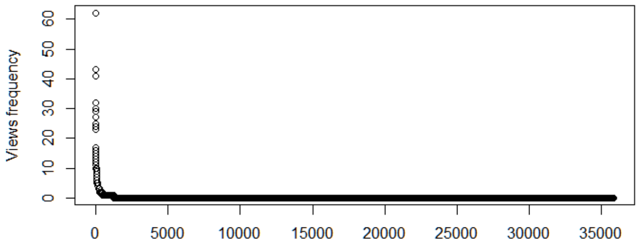


Figure 4: Views frequency for data.gouv.fr datasets

The analysis of percentile values is more interesting. They have a very similar trend: in any case we have that 25% of datasets is not nearly used, i.e. 0 downloads for UK and just 6 views for about 12,000 datasets; just 1 views for about 60,000 dataset for the U.S. portal; 0 views for both the Irish and French portals. Things are just getting better if we look at the median: especially in the UK portal other 12,000 datasets are visualized at most by 26 users, of whom at most 4 have downloaded the viewed datasets. The data for U.S. portal bear witness to lower values with about half of the datasets (118,000) viewed by no more than 12 users. For the IE portal, half of the datasets (4,500) have been viewed no more than three times. While the French portal show that half its datasets have not be viewed. When looking at the 3rd quartile these values slightly improve. The English portal recorded 104 views and 19 downloads for at least 12,000 datasets, while 25% of data.gov datasets have been viewed by at least 19 users, similar to the numbers recorded

[11] https://data.gov.uk/data/site-usage/dataset - retrieved January 16, 2019

Table 2: Statistics of usage metrics for the US, UK, IE, F and HDX portals

Portal	Metric	Min.	1st. Qu.	Median	Mean	3rd Qu.	Max
data.gov	Views	0	1	12	34	19	127,643
data.gov.uk	Views	1	6	26	291	104	204,803
	Downloads	0	0	4	79	19	139479
data.gov.ie	Views	0	0	3	56	20	17,248
data.gouv.fr	Views	0	0	0	76	1	160,003
data.human.org	Views	0	0	0	3.3	2	444
	Downloads	0	1	17	168	96	13,309

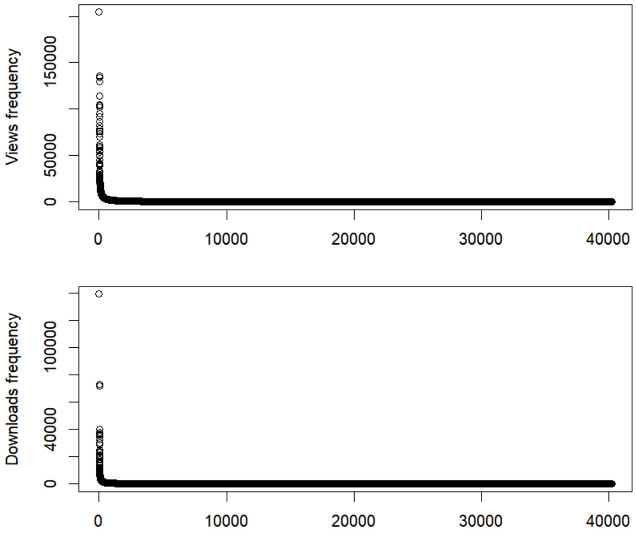


Figure 5: Views and downloads frequency for data.gov.uk datasets

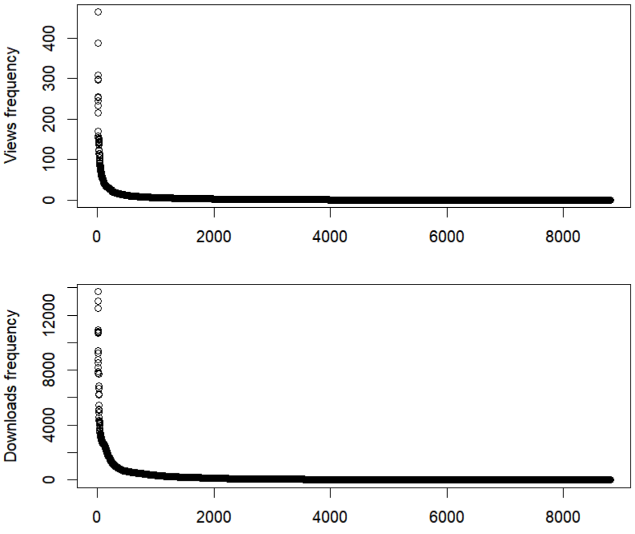


Figure 6: Views and downloads frequency for data.humdata.org datasets

for the IE portal. As for HDX, considering that the 14-day views sample provides rather modest results, it seems more significant the number of those who downloaded the datasets with about half of them downloaded at least 17 times and a quarter at least one hundred.

To supply an approximate idea of the frequency ranges of the most used ones, we reported in Table 3 the 90, 95 and 99 percentiles for each portal and each metric available. In the case of U.S. the under-usage rates seem partly confirmed also for 90 and 95 percentiles, i.e. 95% datasets have been viewed no more than 59 times. Just 1% (about 2,300) datasets have been viewed at least by 298 users. UK portal shows better performance: 10% datasets have been viewed at least 358 times and been downloaded 75 times, while about 400 (i.e. the 99% percentile) have about 4,000 views and 1,000 downloads. The IE portal lays in the middle, recording about 900 datasets viewed almost 100 times and about 90 viewed by one thousand users. The French usage figures are still the lowest among the national portals, even in the case of the highest percentiles. If the values of views (but at 14 days) for HDX are relatively significant for the higher percentiles, those of absolute downloads are more than encouraging, with 10% of the datasets seen at least (approximately) 400 times, culminating with more than 2600 downloads for 80 'top' datasets.

4 DISCUSSION

Our study highlights two issues related to usage of OGD portals: i) they are largely underused; ii) they generally do not provide users with usage information.

Table 3: Percentiles of usage metrics for the U.S., UK, IE, F and HDX portals

Portal	Metric	90%	%95	%99
data.gov	Views	33	59	298
data.gov.uk	Views	358	807	3,959
	Downloads	73	178	1,071
data.gov.ie	Views	98	233	998
data.gouv.fr	Views	4	10	302
data.human.org	Views	7	13	46
	Downloads	376	655	2,620

4.1 Underutilisation of OGD datasets

The results of our analysis, albeit partial because of the small number of considered OGD portals, highlight a situation that seems common to portals with different dimensions and missions: the majority of the published datasets is used marginally. This seems to confirm the 'fears' expressed by the CDOs survey presented in [17], mentioned in the Introduction.

Obviously, the direct use observable through the adopted metrics does not exhaust the potential of the data offered by the portals: as mentioned, probably a more meaningful parameter is tied to the number of indirect users, namely those that use third-party applications, in combination with the number of the same applications. If the number of users of an application can be difficult to collect and assign to a dataset, the number of applications using a data set could be collected as done for the aforementioned French and Portuguese portals. This can improve the perception of the utility of a dataset and provide an indicator for a quantitative assessment of the indirect use of the portal. To facilitate this recognition, third-party applications that use a data set should always be encouraged to list it among their sources[21]. This would help users not only to know the provenance of the original data, but even more to make the products of these applications reproducible and therefore more reliable[1].

Although the choice of metrics we adopted can influence the extent of the assessments on the use of the dataset, they provide significant indicators to the CDOs to understand if the datasets published on the portals they manage attract the interest of the users. According to one of them: "We look at the total number of datasets that are out there, what we are offering up. We count visit clicks, and lastly, we look at how many downloads are actually being done off the open data portal" [17].

One wonders then why some some datasets get more attention than others do, and in some cases thousands of datasets are completely 'invisible' to users. A plausible cause is attributable to the degree of popularity of the thematic domain of each dataset. As emerged from the cited European Community report [3], some thematic domains (e.g. Government and public sector, Population and social conditions) are more popular than other (e.g. Health, Justice and Public safety). We focused on UK portal and we verified the impact of the thematic domains on the data Views: we analyzed the thematic domains of the most viewed datasets (i.e., belonging to 95 percentile) and the less viewed (i.e. belonging to 25th percentile). The aims is to understand if some thematic domain turn out to be the prerogative of the most viewed datasets. Examining the two graphs in figures 7 and 8 this hypothesis seems to apply in particular to the datasets cataloged with respect to the thematic domains 'society' and 'health' which in a significant percentage belong to the 95 percentile, while to a lesser extent to the 25 percentile. Conversely, the datasets of the 'environment' thematic domain, although they are among the most present in the 95 percentile, with about 16% of the most viewed, are also those with more presences (40%) in the lower quartile. According to Figure 9 the 'environment' datasets are also the most present (i.e. 26%) in the UK portal.

Since membership of a certain thematic domains does not seem to be entirely relevant to the fact that some datasets are more used than others are, other complementary causes must be sought. Based on the literature we believe it is worth investigating whether there is any correlation between the popularity of a dataset and the quality of its data [22] and metadata [9]. While, the quality of the dataset can only be analyzed when it has been downloaded in whole or in part, and therefore the fact that it is not re-used is only an ex-post consequence of its usage, the quality of the metadata can effectively preclude visibility (to search engines, even inside OGD portals) and therefore immediate use [8][12]. In addition, users may be disoriented by the non adoption of metadata standards or by their heterogeneity, amongst different portals. To face these issues initiatives (such as W3C[12], OGC[13], INSPIRE[14], FAIR[15]) recommend providing metadata according to existing standards, to "facilitate interoperability between data catalogues published on Web". In

[12] https://www.w3.org/
[13] https://www.opengeospatial.org/standards
[14] https://inspire.ec.europa.eu/
[15] https://www.go-fair.org/fair-principles/

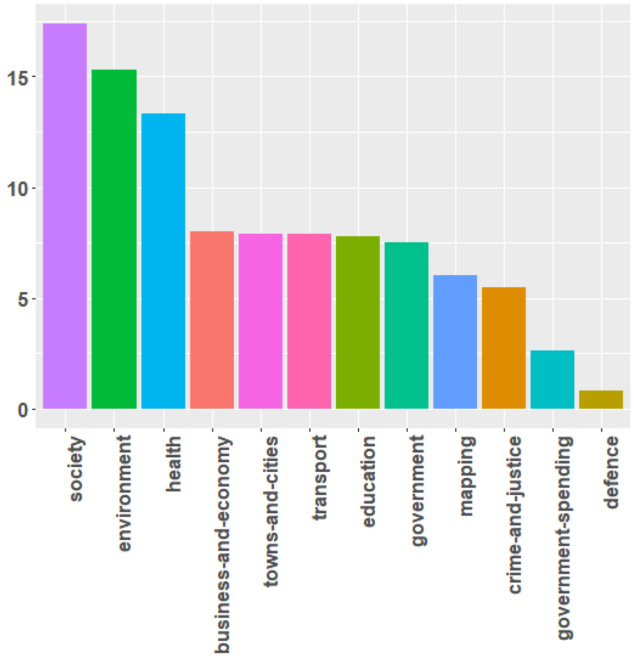


Figure 7: Distribution of the Views for the top (95 percentile) datasets on data.gov.uk according to thematic domains.

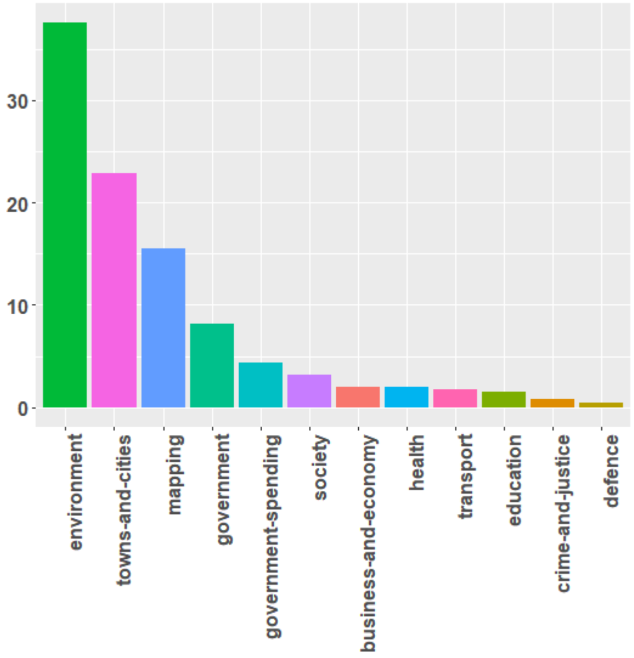


Figure 8: Distribution of the Views for the bottom (25 percentile) datasets on data.gov.uk according to thematic domains.

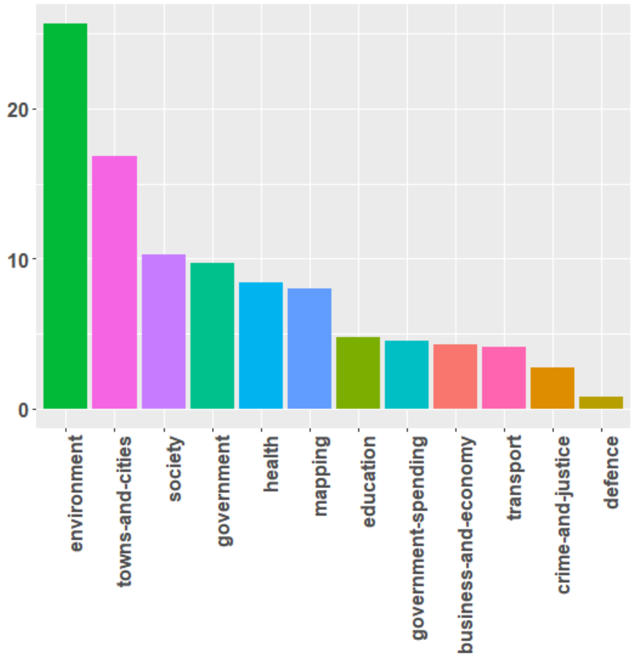


Figure 9: Distribution of popularity (according to Views) of datasets of the UK portal respect to thematic domains.

particular, DCAT[16] has recently designed to improve the data catalogues interoperability and to allow applications to easily consume metadata from multiple catalogues[17].

4.2 Scarcity of usage data

A second critical aspect, which emerged from our analysis, concerns the rarity or non-availability of usage information. As stated in Sections 2.1 and 3.1, in deciding which OGD portals to include in our analysis we realized the difficulty in finding any usage information, both direct and indirect, in most of the main OGD portals managed at national or international level. In addition, in those few portals where this information is made available, the usage indicators are not always all present (see Table 1). Or again, as in the portals in which we found usage metadata available directly to the users, the programmatic access enables to find only partial or empty metadata as highlighted for the French, Portuguese and UK portals. Hence the difficulty of obtaining a homogeneous synthesis able to provide a systematic image on the use of the different portals. This lack would seem to imply that CDOs underestimate the importance of informing users about the popularity of their datasets. As observed by Sasse [16] however, use indicators such as Views and Downloads would have the potential effect of diverting users’ attention to the datasets published on their portal, instead of those available on competing portals. In short, this ‘popularity’ information could work similarly to that used to attract users / customers to a social media or web economy platform and be collected by government

[16] https://www.w3.org/TR/vocabdcat/

[17] http://devinit.org/wp-content/uploads/2018/01/Metadata-for-open-data-portals.pdf

portals thus to improve customer service [5]. The fact that this does not happen, in most of the OGD portals, makes us suspect that in such cases the effect "no one visits it", as was the case for many of the datasets examined, would constitute a "boomerang effect" that the managers of the portals prefer to avoid[18].

5 CONCLUSIONS AND FUTURE WORKS

The paper provides a preliminary overview about the use of OGD portals. It contributes i) to outline a common under-use of most (of the datasets) of OGD portals; ii) to highlight the lack of usage (meta)data of the datasets by the portals themselves.

Regarding the first point, from a set of 98 national OGD portals around the world, we have selected a subset of five which provide direct (metadata) or indirect (via access via programmatic API) access to the usage metrics (i.e. Views and Download) in each portal. The results show that the frequencies of use follow a long-tailed distribution for all the portals analyzed. From a first investigation, it seems that the reason for a preference, so relevant to users for few datasets, is not immediately attributable to their belonging to a particular thematic domain. We advance instead the hypothesis that this is in some way related to the quality of their metadata.

As for the second point, we have given some insights on practices of publication of datasets usage metadata by portal operators, noting that usage metrics values such as the numbers of views and downloads are not easily accessible to users or missing from most of OGD portals. As observed this makes less immediate for users to evaluate the reception of a dataset of interest to them.

In our future work, we will analyze the potential correlation between the popularity of the data set and the quality of its data/metadata, considering that, as recommended by the literature, it is advisable to publish good quality data and metadata to improve user understanding and, therefore, increase the use of the associated datasets.

REFERENCES

[1] Riccardo Albertoni, Monica De Martino, and Alfonso Quarati. 2018. Documenting Context-based Quality Assessment of Controlled Vocabularies. *IEEE Transactions on Emerging Topics in Computing* 01 (jan 2018), 1–1. https://doi.org/10.1109/TETC.2018.2865094

[2] Jorn Berends, Wendy Carrara, and Cosmina Radu. 2017. *The Economic Benefits of Open Data.* Analytical Report 9.

[3] Gianfranco Cecconi and Cosmina Radu. 2018. *Open Data Maturity in Europe 2018.* https://www.europeandataportal.eu/sites/default/files/edp_landscaping_insight_report_n4_2018.pdf

[4] Christian Philipp Geiger and Jörn Von Lucke. 2012. Open government and (linked)(open)(government)(data). *JeDEM-eJournal of eDemocracy and open Government* 4, 2 (2012), 265–278.

[5] R. C. Joseph and N. A. Johnson. 2013. Big Data and Transformational Government. *IT Professional* 15, 6 (Nov 2013), 43–48. https://doi.org/10.1109/MITP.2013.61

[6] Rashmi Krishnamurthy and Yukika Awazu. 2016. Liberating data for public value: The case of Data.gov. *International Journal of Information Management* 36, 4 (2016), 668 – 672. https://doi.org/10.1016/j.ijinfomgt.2016.03.002

[7] Rui Pedro Lourenço. 2015. An analysis of open government portals: A perspective of transparency for accountability. *Government Information Quarterly* 32, 3 (2015), 323 – 332. https://doi.org/10.1016/j.giq.2015.05.006

[8] Renata Machova and Martin Lnenicka. 2017. Evaluating the Quality of Open Data Portals on the National Level. *Journal of theoretical and applied electronic commerce research* 12 (00 2017), 21 – 41.

[9] Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. 2016. Automated Quality Assessment of Metadata Across Open Data Portals. *J. Data and Information Quality* 8, 1, Article 2 (Oct. 2016), 29 pages. https://doi.org/10.1145/2964909

[10] OECD. 2018. *Open Government Data Report: Enhancing Policy Maturity for Sustainable Impact.* OECD Publishing, Paris. 264 pages. https://doi.org/10.1787/9789264305847-en

[11] Monica Palmirani, Michele Martoni, and Dino Girardi. 2014. Open Government Data Beyond Transparency. In *Electronic Government and the Information Systems Perspective*, Andrea Kő and Enrico Francesconi (Eds.). Springer International Publishing, Cham, 275–291.

[12] K.J. Reiche and E. Hofig. 2013. Implementation of metadata quality metrics and application on public government data. *Proceedings - International Computer Software and Applications Conference*, 236–241. https://doi.org/10.1109/COMPSACW.2013.32

[13] Sergio Rosim, Larcio Massaru Namikawa, Joo Ricardo de Freitas Oliveira, Monica De Martino, and Alfonso Quarati. 2018. Workflow Provenance Metadata to Enhance Reuse of South America Drainage Datasets. In *2018 International Conference on eDemocracy eGovernment (ICEDEG)*. 16–23. https://doi.org/10.1109/ICEDEG.2018.8372337

[14] Shazia Sadiq and Marta Indulska. 2017. Open data: Quality over quantity. *International Journal of Information Management* 37, 3 (2017), 150–154.

[15] Igbal Safarov, Albert Jacob Meijer, and Stephan Grimmelikhuijsen. 2017. Utilization of open government data: A systematic literature review of types, conditions, effects and users. *Information Polity* 22 (2017), 1–24.

[16] Tom Sasse, Amanda Smith, Ellen Broad, Jeni Tennison, Peter Wells, and Ulrich Atz. 2017. Recommendations for Open Data Portals: from setup to sustainability. https://www.europeandataportal.eu/sites/default/files/edp_s3wp4_sustainability_recommendations.pdf

[17] Adam Stone. 2018. Are Open Data Efforts Working? *government technology* (Mar 2018). http://www.govtech.com/data/Are-Open-Data-Efforts-Working.html

[18] Barbara Ubaldi. 2013. Open Government Data. 22 (2013). https://doi.org/https://doi.org/10.1787/5k46bj4f03s7-en

[19] Sander van der Waal, Krzysztof Węcel, Ivan Ermilov, Valentina Janev, Uroš Milošević, and Mark Wainwright. 2014. *Lifting Open Data Portals to the Data Web.* Springer International Publishing, Cham, 175–195. https://doi.org/10.1007/978-3-319-09846-3_9

[20] Francois Van Schalkwyk and Stefaan G Verhulst. 2017. The state of open data and open data research. https://doi.org/10.5281/zenodo.1117807 Published by African Minds.

[21] Miel Vander Sande, Marc Portier, Erik Mannens, and Rik Van de Walle. 2012. Challenges for open data usage: open derivatives and licensing. In *Workshop on Using Open Data, Proceedings*. 4.

[22] Antonio Vetró, Lorenzo Canova, Marco Torchiano, Camilo Orozco Minotas, Raimondo Iemma, and Federico Morando. 2016. Open data quality measurement framework: Definition and application to Open Government Data. *Government Information Quarterly* 33, 2 (2016), 325 – 337. https://doi.org/10.1016/j.giq.2016.02.001

[23] Anneke Zuiderwijk, Marijn Janssen, and Iryna Susha. 2016. Improving the speed and ease of open data use through metadata, interaction mechanisms, and quality indicators. *Journal of Organizational Computing and Electronic Commerce* 26, 1-2 (2016), 116–146. https://doi.org/10.1080/10919392.2015.1125180

[18] https://www.forbes.com/sites/ryanerskine/2017/09/19/20-online-reputation-statistics-that-every-business-owner-needs-to-know

Simplified Data Posting in Practice

Elio Masciari
ICAR-CNR
Rende (CS), Italy
elio.masciari@icar.cnr.it

Irina Trubitsyna
University of Calabria
Rende (CS), Italy
trubitsyna@dimes.unical.it

Domenico Saccà
University of Calabria
Rende (CS), Italy
sacca@dimes.unical.it

ABSTRACT

The data posting framework introduced in [8] adapts the well-known Data Exchange techniques to the new Big Data management and analysis challenges that can be found in real world scenarios. Although it is expressive enough, it requires the ability of using count constraints and may be difficult for a non expert user. Moreover, the data posting problem is *NP*-complete under the data complexity in the general case, then the use of the non-deterministic variables is performed. Indeed, identifying the conditions that guarantee polynomial-time execution in the presence of non-deterministic choices is very important for practical purposes. In this paper, we present a simplified version of data posting framework, based on the use of the *smart mapping rules*, that integrate the simple mapping description with some parameters, avoiding the complex specifications with count constraints. We show that the data posting problem in the new setting is *NP*- complete and identify the conditions under which this problem becomes polynomial even in the presence of non-deterministic choices.

CCS CONCEPTS

• **Applied computing** → **Document searching**;

KEYWORDS

Data Posting, Big Data

ACM Reference Format:
Elio Masciari, Irina Trubitsyna, and Domenico Saccà. 2019. Simplified Data Posting in Practice. In *IDEAS 2019: 23rd International Database Engineering & Applications Symposium, June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3216122.3216154

1 INTRODUCTION

Big Data paradigm[1, 32, 33] recently come on scene in a quite pervasive manner, however this apparently sudden change of perspective had a long history before the term *Big Data* was coined. Indeed, both industry and research people have been entrenched in (big) data that have been stored in massive amounts, with an increasing speed and exhibiting a huge variety for over a decade before the Big Data paradigm was *officially* born. The major challenge has always been to unveil valuable insights for the industry to which these particular data belonged.

As a matter of fact, sifting through all of these data, parsing them, transferring them from a source to a target database, and analyzing all of them for purposes of improving business decision-making processes turn to be too complex for traditional approaches. In the presence of incomplete databases, certain answers are a principled semantics of query answering [10, 15]. Since the computation of certain query answers is a coNP-hard problem, recent research has focused on developing polynomial time algorithms computing a sound (but possibly incomplete) set of certain answers [12, 16, 17, 22, 23]. Approximation algorithms offers a possible solution, when detailed information is required. However, in the Big Data scenario we are often interested in succinct information and in the discovery of new knowledge. To address this issues, some proposals have been made, like the *Data Posting* framework [8]. One of the most important features of Data Posting is the enriching data while exchanging them between the sources and the target database. Intuitively, the Data Posting setting consists of a source and a domain database schemes, a target flat fact table, a set of source-to-target mapping rules and a set of target constraints. The *data posting problem* associated with this setting is: given finite source and domain database instances, find a finite instance for the target fact table that satisfies the internal integrity constraints and the mapping requirements.

The problem of finiteness of the Target database is well known in the context of Data Exchange. The presence of existential quantifiers in the mapping rules and their replacement with null values can create situations in which the finiteness property of the Target database could be not satisfied. Data Posting approach use non-deterministic variables instead of the existentially quantified ones in the mapping rules (the so called, Source to Target Generating Dependencies). The values for the non-deterministic variables can be chosen non deterministically from the finite domain relation following a strategy that leverages count constraints [31]. Obviously, the solution to the data posting problem could not be universal as it represents a specific choice. However, as mentioned before, in the context of Big Data we are often interested in the discovery of new knowledge and the overall analysis of the data, moreover some attributes of the target tables may be created for storing the discovered values. Thus, the choice of actual values can be seen as a first phase of data analysis that solves uncertainties by enriching the information contents of the whole system. Consider the following application scenario that we will use as running example.

EXAMPLE 1. *The databases S_1 and S_2 describe the user's profiles represented by relations* $\mathtt{P}_1(\mathtt{I},\mathtt{N},\mathtt{V})$ *and* $\mathtt{P}_2(\mathtt{I},\mathtt{N},\mathtt{V})$ *respectively, with attributes* I *(profile's identifier),* N *(attribute's name) and* V *(attribute's value). The problem is to enrich the user profiles from* $\mathtt{P}_1$ *with some "relevant" attributes from* $\mathtt{P}_2$. □

IDEAS 2019, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8…$15.00
https://doi.org/10.1145/3216122.3216154

The scenario described in the example above requires the solution of two main problems:

(1) extract the information about profiles compatibility in the relations $\mathtt{P}_1$ and $\mathtt{P}_2$
(2) identify "relevant" attributes and their values for each profile in $\mathtt{P}_1$.

The first task can be considered as a kind of soft-clustering that aims at grouping similar users in order to classify them for further analysis, e.g., mail classification [24], trajectory grouping[25], biological data analysis[26]. The second one requires the definition of the choice strategy based on the user/designer experience. In particular, the choice of the name-value combination has to take into account two different needs: 1) deciding if the attribute has to be added to the profile, 2) selecting the value of this attribute like for classical data warehousing [11]. One possible criterion and its expression with the standard data posting constructs is described below.

Example 1 (continued). Suppose that we extract the information about profiles compatibility and store them into a relation $\mathtt{C}(\mathtt{I}_1, \mathtt{I}_2, \mathtt{L})$. In particular, the first two attributes contain profile's identifiers from tables $\mathtt{P}_1$ and $\mathtt{P}_2$, respectively, whereas $\mathtt{L}$ represents the level of compatibility of these profiles. In order to enrich $\mathtt{P}_1$ with some "relevant" attributes from $\mathtt{P}_2$ we can set the following strategy: *An attribute combination name-value (n_2, v_2) taken from* $\mathtt{P}_2$ *is "relevant" to the profile with identifier I_1 described in the relation* $\mathtt{P}_1$ *if the following conditions hold: 1) it is* sufficiently supported*, i.e. it is* supported *by at least* 10 *profiles from* $\mathtt{P}_2$ *with a percentage of compatibility towards i_1 at least* 50%*; 2) if different values corresponding to the same attribute are sufficiently supported, only the one with the greatest support value is "relevant" and will be added to I_1.*

The description of this scenario with the standard data posting constructs can be done as follows. We define a domain relation containing only the values 0, 1, and −1. The target relations are:

- $\mathtt{A}(\mathtt{I}_1, \mathtt{I}_2, \mathtt{N}_2, \mathtt{V}_2)$ stores the information of the profile's from $\mathtt{P}_2$, whose compatibility level with some profile in $\mathtt{P}_1$ is at least 50%.
- $\mathtt{Add}(\mathtt{I}_1, \mathtt{N}_2, \mathtt{V}_2, \mathtt{Flag})$ stores the combinations name-value $\langle N_2, V_2\rangle$ taken from $\mathtt{P}_2$ and the decision $\mathtt{Flag}$ to add this couple to the profile $\mathtt{I}_1$. The values of the attribute $\mathtt{Flag}$ have the following meaning:
  -1 the combination $\langle N_2, V_2\rangle$ is not added, because it is not sufficiently supported;
  0 the combination $\langle N_2, V_2\rangle$ is not added, although it is sufficiently supported, the value V_2 is not selected following the preference specification;
  1 the combination $\langle N_2, V_2\rangle$ is added, as it is sufficiently supported and the value V_2 is selected following the preference specification.

The source to target dependencies are

$$\mathtt{P}_2(\mathtt{i}_2, \mathtt{n}_2, \mathtt{v}_2) \wedge \mathtt{C}(\mathtt{i}_1, \mathtt{i}_2, \mathtt{l}) \wedge \mathtt{l} \geq 0,50 \longrightarrow \mathtt{A}(\mathtt{i}_1, \mathtt{i}_2, \mathtt{n}_2, \mathtt{v}_2)$$
$$\mathtt{P}_2(\mathtt{i}_2, \mathtt{n}_2, \mathtt{v}_2) \wedge \mathtt{C}(\mathtt{i}_1, \mathtt{i}_2, \mathtt{l}) \wedge \mathtt{l} \geq 0,50 \wedge \mathcal{D}(\mathtt{flag}) \longrightarrow \mathtt{Add}(\mathtt{i}_1, \mathtt{n}_2, \mathtt{v}_2, \mathtt{flag})$$

where all variables are universally quantified. The fact that $\mathcal{D}$ is domain relation and its presence in the body of the second constraint states that only one value between −1, 0 and 1 can be chosen as the $\mathtt{flag}$ value for each triple $(\mathtt{i}_1, \mathtt{n}_2, \mathtt{v}_2)$ in the relation $\mathtt{Add}$.
The following count constrains set the selection criteria:

$$\mathtt{Add}(\mathtt{i}_1, \mathtt{n}_2, \mathtt{v}_2, 1) \longrightarrow \#(\{\, \mathtt{I}_2 : \mathtt{A}(\mathtt{i}_1, \mathtt{I}_2, \mathtt{n}_2, \mathtt{v}_2)\}) \geq 10$$
$$\mathtt{Add}(\mathtt{i}_1, \mathtt{n}_2, \mathtt{v}_2, 0) \longrightarrow \#(\{\, \mathtt{I}_2 : \mathtt{A}(\mathtt{i}_1, \mathtt{I}_2, \mathtt{n}_2, \mathtt{v}_2)\}) \geq 10$$
$$\mathtt{Add}(\mathtt{i}_1, \mathtt{n}_2, \mathtt{v}_2, -1) \longrightarrow \#(\{\, \mathtt{I}_2 : \mathtt{A}(\mathtt{i}_1, \mathtt{I}_2, \mathtt{n}_2, \mathtt{v}_2)\}) < 10$$
$$\mathtt{Add}(\mathtt{i}_1, \mathtt{n}_2, \_, \_) \longrightarrow \#(\{\, \mathtt{V} : \mathtt{Add}(\mathtt{i}_1, \mathtt{n}_2, \mathtt{V}, 1)\}) = 1$$
$$\mathtt{Add}(\mathtt{i}_1, \mathtt{n}_2, \mathtt{v}_2, 1), \mathtt{Add}(\mathtt{i}_1, \mathtt{n}_2, \mathtt{v}, 0) \longrightarrow \#(\{\, \mathtt{I}_2 : \mathtt{A}(\mathtt{i}_1, \mathtt{I}_2, \mathtt{n}_2, \mathtt{v}_2)\}) \geq \#(\{\, \mathtt{I}_2 : \mathtt{A}(\mathtt{i}_1, \mathtt{I}_2, \mathtt{n}_2, \mathtt{v})\})$$

□

The data posting setting is expressive enough, however, as shown in the example above, it requires the ability of using count constraints and may be difficult for a non expert user. Moreover, as shown in [30] the complexity of the data posting problem is *NP*-complete under the data complexity in the general framework, where the use of the non-deterministic variables is performed. In the absence of non-deterministic variables, the problems becomes polynomial, but this condition is too restrictive in practice. Thus, identifying the conditions that guarantee polynomial-time execution in the presence of non-deterministic choices is very important for practical purposes.

Recently, in [27], the use of *smart mapping rules* to support user suggestion in a big data environment has been proposed. In this paper we further investigate this idea and present a simplified version of data posting framework, called Smart Data Posting. We show that the data posting problem in the new setting is *NP*-complete and identify the conditions when this problem becomes polynomial even in the presence of non-deterministic choices.

The model of our running scenario in the new framework is described below.

Example1 (continued). Our running scenario can be modelled as follows.

- $\mathtt{P}_2$ and $\mathtt{C}$ are source relations;
- $\mathtt{Add}(\mathtt{I}_1, \mathtt{N}_2, \mathtt{V}_2)$ is target relation, which specifies the tuples to be added in the other target relation $\mathtt{P}_1$;
- the smart mapping rule is reported below:

$$\mathtt{P}_2(\mathtt{i}_2, \mathtt{n}_2, \mathtt{v}_2) \wedge \mathtt{C}(\mathtt{i}_1, \mathtt{i}_2, \mathtt{l}) \wedge \mathtt{l} \geq 0,50 \xrightarrow{\mathtt{i}_2, 10, \langle \mathtt{v}_2, \mathtt{unique} \wedge \mathtt{max}\rangle} \mathtt{Add}(\mathtt{i}_1, \mathtt{n}_2, \mathtt{v}_2)$$

Intuitively, the body of the rule allows to restrict the attention to the profiles with at least 50% compatibility, the selection criterion has been synthesized on the arrow, indicating 1) the support variable (i_2), 2) the minimum quantity (10) of support instances to be able to map, and 3) the variable (v_2) whose value should be chosen from the set of candidate values with their preference for this choice (*unique* ∧ *max*). □

Note that the use of the smart mapping rule makes the model of the described scenario more simple and intuitive. Indeed, this representation evidences, that the analyst can built the local selection criterion focusing only on a small set of parameters. The selection criteria, that can be represented by smart mapping rules, is a particular parametric combination of aggregation, counting and selection operations. The standardization of its representation allows us to optimize the implementation of the data posting process.

Plan of the paper. In Section 2 we describe the background of our approach. In Section 3 we present the Smart Data Posting framework. In Section 4 we perform a complexity analysis of the framework. Finally, in Section 5 we draw our conclusion.

2 BACKGROUND

Data Exchange. A schema is a finite collection $R = \{R_1, ..., R_k\}$ of relation symbols. Each relation symbol has an arity, which is a positive integer. A relation symbol of arity n is called n-ary, and has n distinct attributes, which intuitively correspond to column names. An instance I over the schema R is a function that associates to each n-ary relation symbol R_i an n-ary relation $I(R_i)$. With a little abuse of notation we will use R_i to denote both the relation symbol and the relation that interprets it. Given a tuple t occurring in a relation R, we denote by $R(t)$ the association between t and R and call it a fact. An instance can be conveniently represented by its set of facts. $R(\mathbf{v})$, where $\mathbf{v}$ is a vector of variables or constants with the arity of R, is called atom. If R is a schema, then a dependency over R is a sentence in some logical formalism over R.

A *Tuple Generating Dependency (TGD)* is formula of the form:

$$\forall \mathbf{x}\, \phi(\mathbf{x}) \rightarrow \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})$$

where $\phi(\mathbf{x})$ and $\psi(\mathbf{x}, \mathbf{y})$ are conjunctions of atoms, and $\mathbf{x}, \mathbf{y}$ are lists of variables.

Full TGDs are TGDs without existentially quantified variables.

An *Equality Generating Dependency (EGD)* is formula of the form:

$$\forall \mathbf{x}\, \phi(\mathbf{x}) \rightarrow x_1 = x_2$$

where $\phi(\mathbf{x})$ is conjunction of atoms, while x_1 and x_2 are variables in $\mathbf{x}$. In our formulae it is common to omit the universal quantifiers, when their presence is clear from the context. The left hand side (w.r.t. the implication symbol) of a data dependency is called *body*, whereas the right hand side is called *head*.

Let $S = S_1, ..., S_n$ and $T = T_1, ..., T_m$ be two disjoint schemas. We refer to S as the source schema and to the S_i's as the source relation symbols. We refer to T as the target schema and to the T_j's as the target relation symbols. Similarly, instances over S will be called source instances, while instances over T will be called target instances. If I is a source instance and J is a target instance, then we write $\langle I, J\rangle$ for the instance K over the schema $S \cup T$ such that $K(S_i) = I(S_i)$ and $K(T_j) = J(T_j)$, for $i \leq n$ and $j \leq m$.

The data exchange setting [2, 10] is a tuple $(S, T, \Sigma_{ST}, \Sigma_T)$, where S is the source relational database schema, T is the target schema, Σ_T are dependencies over the target schema T and Σ_{ST} are source-to-target TGDs. The dependencies in Σ_{ST} map data from the source to the target schema and are TGDs of the form

$$\forall \mathbf{x}(\ \phi_S(\mathbf{x}) \rightarrow \exists \mathbf{y}\ \psi_T(\mathbf{x}, \mathbf{y})\ )$$

where $\phi_S(\mathbf{x})$ and $\psi_T(\mathbf{x}, \mathbf{y})$ are conjunctions of atomic formulas on S and T, respectively. Dependencies in Σ_{ST} are also called mapping dependencies. Dependencies in Σ_T specify constraints on the target schema and can be either TGDs or EGDs.

The data exchange problem associated with this setting is the following: given a finite source instance I, find a finite target instance J such that $\langle I, J\rangle$ satisfies Σ_{ST} and J satisfies Σ_T. Such a J is called a solution for I.

The computation of an universal solution (the compact representation of all possible solutions) can be done by means of the fixpoint chase algorithm, when it terminates [9]. The execution of the chase involves inserting tuples possibly with null values to satisfy TGDs, and replacing null values with constants or other null values to satisfy EGDs. Specifically, the chase consists of applying a sequence of steps, where each step enforces a dependency that is not satisfied by the current instance. It might well be the case that multiple dependencies can be enforced and, in this case, the chase picks one nondeterministically. Different choices lead to different sequences, some of which might be terminating, while others might not. Unfortunately, checking whether the chase terminates is an undecidable problem [9]. To cope with this issue, several "termination criteria" have been proposed, that is, (decidable) sufficient conditions ensuring chase termination. Some recent works can be found in [3, 4, 19, 20, 28, 29], a tool for checking chase termination has been described in [13].

Data posting. Differently from the standard Data Exchange approach, the Data Posting [30] search for more expressive constraints to enrich the contents of the exchanged data. We start from the definition of the involved database schemata.

Let $\mathrm{S} = \langle S_1, \ldots, S_n\rangle$, $\mathcal{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_m\rangle$, and $\mathbf{T} = \langle T_1, \ldots, T_q\rangle$ be be two disjoint schemas. We refer to S (resp. $\mathcal{D}$, T) as the *source* (resp. *domain, target*) schema and to the S_i's (resp. $\mathcal{D}_j$, T_k) as the source (resp. *domain, target*) relation symbols. We assume that all instances over S and $\mathcal{D}$ are finite. As it will be shown in this section, any target instance over $\mathbf{T}$ is finite as well, given the structure of our mappping constraints defined below.

A *non-deterministic source-to-target TGD* is a dependency over $\langle \mathrm{S}, \mathcal{D}, \mathbf{T}\rangle$ of the form

$$\forall \mathbf{x}\, [\ \phi_S(\mathbf{x} \cup \tilde{\mathbf{y}}) \rightarrow \phi_T(\mathbf{z})\ ]$$

where $\mathbf{x}$ and $\mathbf{z}$ are lists of universally quantified variables; $\tilde{\mathbf{y}}$ is a (possibly empty) list of variables, called *non deterministic*, these variables can occur in ϕ_S only in relations from $\mathcal{D}$; $\mathbf{x} \cap \tilde{\mathbf{y}} = \emptyset$ and $\mathbf{z} \subseteq \mathbf{x} \cup \tilde{\mathbf{y}}$; the formula ϕ_S and ψ_T are conjunctions of atoms with predicate symbols in $\mathrm{S} \cup \mathcal{D}$ and in $\mathbf{T}$, respectively.

The non-deterministic TGDs can be seen as the standard TGDs there existentially quantified variables are replaced with non-deterministic variables, whose values can be chosen from the finite domains defined by domain relations. The mapping process is performed as usual but presumes that for every assignment of $\mathbf{x}$ a subset of all admissible values for $\tilde{\mathbf{y}}$ can be chosen in an arbitrary way. Every possible choice is called *non-deterministic domain mapping*.

Let $I = (I_S, I_\mathcal{D})$ be given, where I_S and $I_\mathcal{D}$ are finite source instances for S and for $\mathcal{D}$, respectively. The *active domain* AD_I is the set of all values occurring in I_S and $I_\mathcal{D}$. Let an *admissible instance* I_T for $\mathbf{T}$ be also given, that is an instance whose values all occur in AD_I. The semantic of t states whether t is satisfied or not by $\langle I_S, I_\mathcal{D}, I_T\rangle$. The notion of satisfiability is introduced after preliminary fixing one of the possible non-deterministic domain mappings, say f_t.

We say that $\langle I_S, I_\mathcal{D}, I_T\rangle$ satisfies t w.r.t. f_t if for each $\mathbf{X} \in (AD_I)^n$ and for each $\tilde{\mathrm{Y}} \in \mathbf{f_t}(\mathrm{X})$: either $\phi_S(\mathbf{x} \cup \tilde{\mathbf{y}})[\mathbf{x}/\mathrm{X}, \tilde{\mathbf{y}}/\tilde{\mathrm{Y}}]$ is made false by $\langle I_S, I_\mathcal{D}\rangle$ or $\phi_T(\mathbf{z})[\mathbf{z}/(\mathrm{X} \cup \tilde{\mathrm{Y}})_\mathbf{z}]$ is made true by I_T, where the substitution $[\mathbf{x}/\mathrm{X}, \tilde{\mathbf{y}}/\tilde{\mathrm{Y}}]$ assigns the values X and $\tilde{\mathrm{Y}}$ to the corresponding variables in $\mathbf{x}$ and $\tilde{\mathbf{y}}$, respectively, in the formula ϕ_S and it induces a substitution, denoted by $[\mathbf{z}/(\mathbf{X} \cup \tilde{\mathbf{Y}})_\mathbf{z}]$ for the variables of $\mathbf{z}$ in the formula ϕ_T as well, since $\mathbf{z} \subseteq \mathbf{x} \cup \tilde{\mathbf{y}}$ by definition.

Given a set Σ of non-deterministic source-to-target TGD constraints and finite source instances I_S for $\mathbf{S}$, $I_{\mathcal{D}}$ for $\mathcal{D}$ and I_T for $\mathbf{T}$, $\langle I_S, I_{\mathcal{D}}, I_T\rangle$ satisfies Σ if for each $t \in \Sigma$, there exists a non-deterministic domain mapping f_t such that $\langle I_S, I_{\mathcal{D}}, I_T\rangle$ satisfies t w.r.t. f_t.

As an example consider the source relation $\mathtt{objects_S}$ and the domain relation $\mathcal{D}$ reporting all possible characterizations of objects, whose instances are $I_S = \{r\}$, where r denotes a restaurant, and $\mathcal{D} = \{(r, fish), (r, meet), (r, expensive), (r, cheap)\}$. The following non-deterministic TGD can be used to assign characterization to the objects choosing them from the domain relation $\mathcal{D}$ non-deterministically.

$$\mathtt{objects_S(n)} \land \mathcal{D}(\mathtt{n}, \mathtt{v}) \rightarrow \mathtt{description_T(n, v)}$$

This constraint is satisfied by $\langle I_S, I_{\mathcal{D}}, I_{T_1}\rangle$, where $I_{T_1} = \{(r, fish), (r, expensive)\}$ and is not satisfied by $\langle I_S, I_{\mathcal{D}}, I_{T_2}\rangle$, where $I_{T_2} = \{(r, green)\}$.

In the case of empty $\tilde{\mathbf{y}}$, the non-deterministic TGD corresponds to a full TGD and its semantics corresponds to the standard one. For instance,

$$\mathtt{objects_S(n)} \rightarrow \mathtt{object_T(n)}$$

simply creates a copy of the source relation $\mathtt{objects_S}$.

A *count constraint* is a dependency over $\mathbf{T}$ of the form

$$\forall \mathbf{x}\, [\, \phi_T(\mathbf{x}) \rightarrow \#(\{\mathbf{y} : \exists\, \mathbf{z}\, \alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})\}) \text{ <op> } \beta(\mathbf{x})\,]$$

where ϕ_T is a conjunction of atoms with predicate symbol in $\mathbf{T}$, <op> is any of the comparison operators ($=$, $>$, $\geq$, $<$ and $\leq$), $H = \{\mathbf{y} : \exists\, \mathbf{z}\, \alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})\}$ is a *set term*, $\#$ is an interpreted function symbol that computes the cardinality of the (possibly empty) set corresponding to H, $\#(H)$ is *count term*, and $\beta(\mathbf{x})$ is an integer or a variable in $\mathbf{x}$ or another count term with universally quantified variables in $\mathbf{x}$. The two lists $\mathbf{y}$ and $\mathbf{z}$ consist of distinct variables that are also different from the universally quantified variables in $\mathbf{x}$, $\alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is a conjunction of atoms $T_i(\mathbf{x}, \mathbf{y}, \mathbf{z})$ with $T_i \in \mathbf{T}$.

To define the semantic of a count constraint, we assume that an instance I_T for $\mathbf{T}$ is given. Then, we consider the *active domain* AD_I as the set of all values occurring in I_T. Given a substitution $\mathbf{x}/\mathbf{X}$ assigning values in AD_I to universally quantified variables, $K_\mathbf{X} = \{\mathbf{y} : \exists\, \mathbf{z}\, \alpha(\mathbf{X}, \mathbf{y}, \mathbf{z})\}$ defines the set of values in AD_I assigned to the free variables in $\mathbf{y}$ for which $\exists\, \mathbf{z}\, \alpha(\mathbf{X}, \mathbf{y}, \mathbf{z})$ is satisfied by I_T and $\#(K_\mathbf{X})$ is the cardinality of this set. We say that I_T satisfies

$$\phi_T(\mathbf{x}) \rightarrow \#(\{\mathbf{y} : \exists\, \mathbf{z}\, \alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})\}) \text{ <op> } \beta(\mathbf{x})$$

if each substitution $\mathbf{x}/\mathbf{X}$ that makes true $\phi_T(\mathbf{x})$, makes also true the head expression $\#(\{\mathbf{y} : \exists\, \mathbf{z}\, \alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})\})$ <op> $\beta(\mathbf{x})$).

As an example, the count constraint

$$\mathtt{object_T(n)} \rightarrow \#(\{\, \mathtt{V} : \mathtt{description(n, V)}\}) = 2$$

states that every object must have exactly 2 characterizations.

Observe that target count constraints extend both TGDs and EGDs of the classical data exchange setting.

We are now ready to formulate the data posting problem:

The *data posting setting* $(\mathbf{S}, \mathcal{D}, T, \Sigma_{ST}, \Sigma_T)$ consists of a source database schema $\mathbf{S}$, a domain database scheme $\mathcal{D}$, a target flat fact table T, a set Σ_{ST} of source-to-target TGDs and a set Σ_T of target count constraints.

The *data posting problem* associated with this setting is: given finite source instances I_S for $\mathbf{S}$ and $I_{\mathcal{D}}$ for $\mathcal{D}$, find a finite instance I_T for T such that $\langle I_S, I_{\mathcal{D}}, I_T\rangle$ satisfies both Σ_{ST} and Σ_T. This problem is NP-complete under the data complexity. Obviously, in the case than Σ_{ST} is composed by only full TGDs, the problem becomes polynomial.

3 SMART DATA POSTING

The Smart Data Posting setting is based on the idea that the standard source to target dependencies can be enriched with the selection criterion regarding the local exchange process. The obtained dependencies, that we call *smart mapping rules*, are expressive enough for different practical applications and can be profitably used for simplifying and optimizing the standard Data Posting setting.

Definition 1. A *smart mapping rule* can be defined as follows:

$$\forall \mathbf{z}[\ \phi(\mathbf{z}) \xrightarrow{\mathbf{y}, k, \langle \mathbf{v}, f\rangle} r(\mathbf{x}, \mathbf{v})\ ]$$

where $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{v}$ are vectors of variables, such that $\mathbf{x} \cup \mathbf{y} \cup \mathbf{v} \subseteq \mathbf{z}$ and $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{v}$ do not share variables; ϕ_S is the conjunction of literals and expressions involving comparison operators ($>, <, \geq, \leq, =, \neq$) and variables in $\mathbf{z}$ or constants; r is a target relation; $\mathbf{y}$ is called a *support vector*; k is a natural number (greater than 1) which indicates the support value; $\mathbf{y}$ and k may be omitted together, the pair $\langle \mathbf{v}, f\rangle$ indicates how the choice for the values of $\mathbf{v}$ should be performed: f can be *max*, *unique*, or the conjunction *unique* $\land$ *max*, the pair $\langle \mathbf{v}, f\rangle$ may be omitted. □

Semantics. The smart mapping rule specifies that the tuple $\langle \mathbf{X}, \mathbf{V}\rangle$ is added to r only if it is supported by at least k (different) initializations $\{\mathbf{Y_1}, ...\mathbf{Y_k}\}$ of $\mathbf{y}$, i.e. for each $j \in [1..k]$ there exists an initialization $\mathbf{Z_j}$ of $\mathbf{z}$, that maps $\mathbf{x}$, $\mathbf{y}$ e $\mathbf{v}$ in $\mathbf{X}$, $\mathbf{Y_j}$ and $\mathbf{V}$ respectively, and that makes true $\phi(\mathbf{Z_j})$. If both $\mathbf{y}$ and k are omitted, all initialization satisfying the body satisfy this first check.

In the case than no further indications of choice are specified (the third arrow label is omitted) all the tuples satisfying the first check are added to r. Otherwise, the set of tuples to be added is further reduced using f for the selection of values in $\mathbf{v}$.

The statement $\langle \mathbf{v}, unique\rangle$ specifies that the tuples transported into the r relation must obey the functional dependency $\mathbf{x} \rightarrow \mathbf{v}$, i.e. for each assignment of values in $\mathbf{x}$ the assignment of values in $\mathbf{v}$ must be unique. In the case than several tuples are supported by at least k (different) initializations of $\mathbf{y}$ and they have the same values in $\mathbf{x}$, the choice can be made arbitrarily.

The statement $\langle \mathbf{v}, max\rangle$ specifies that, for each $\mathbf{X}$ only tuples supported by a maximum number of initializations of $\mathbf{y}$ must be selected. It is easy to see that this constraint does not guarantee the uniqueness of the choice. For example, it is possible that two tuples $\langle \mathbf{X}, \mathbf{V_1}\rangle$ and $\langle \mathbf{X}, \mathbf{V_2}\rangle$ have the same degree of support corresponding to the maximum value.

The statement $\langle \mathbf{v}, unique \land max\rangle$ specifies that, fixed $\mathbf{X}$, only one tuple $\langle \mathbf{X}, \mathbf{V}\rangle$ can be chosen among those supported by a maximum number of (different) initializations of $\mathbf{y}$.

EXAMPLE 2. Consider again our running Example 1. Below we report some selection strategies and the corresponding smart mapping rules.

(1) A description of the attribute's name-value $\langle n_2, v_2 \rangle$ stored in S_2 is added to the profile I_1 only if it is "supported" by at least 10 profiles of the source S_2 with a compatibility percentage towards I_1 of at least 50%. In the case of different combinations characterized by the same name but having the different value, the "most supported" ones are chosen (they can be two or more combinations supported by the same number of profiles in S_2).

$$\mathtt{P_2(i_2, n_2, v_2) \land C(i_1, i_2, l) \land l \geq 0,50} \xrightarrow{\mathtt{i_2, 10, \langle v_2, max \rangle}} \mathtt{Add(i_1, n_2, v_2)}$$

(2) All name-value combinations $\langle n_2, v_2 \rangle$ "supported" by at least 100 profiles belonging to source S_2 with a percentage of compatibility towards profile I_1 of at least 70% must be added to profile I_1.

$$\mathtt{P_2(i_2, n_2, v_2) \land C(i_1, i_2, l) \land l \geq 0,70} \xrightarrow{\mathtt{i_2, 100}} \mathtt{Add(i_1, n_2, v_2)}$$

(3) The attributes to be added to the profile I_1 are those present in the profiles of the source S_2 with a percentage of compatibility towards I_1 of at least 40%. If there are different combinations characterized by the same name having different values, only one combination is chosen arbitrarily.

$$\mathtt{P_2(i_2, n_2, v_2) \land C(i_1, i_2, l) \land l \geq 0,40} \xrightarrow{\mathtt{\langle v_2, unique \rangle}} \mathtt{Add(i_1, n_2, v_2)}$$

□

We will call a smart mapping rule *non-trivial* if the arrow has at least one label, and *trivial* otherwise. Obviously, trivial mapping rules correspond to full TGDs.

DEFINITION 2. *The Smart Data Posting setting (S, T, Σ, Σ_T) consists of a source and a target database schemes S and T, a set , a set Σ of smart mapping rules, and a set Σ_T of target constraints. Smart mapping rules in Σ are of the form $\forall \mathbf{z}[\ \phi_S(\mathbf{z}) \xrightarrow{\mathbf{y}, k, \langle \mathbf{v}, f \rangle} r(\mathbf{x}, \mathbf{v})\ ]$, where ϕ_S denotes the conjunction of source relations. Each target relation defined by the non-trivial mapping rule is defined by only this rule. Σ_T is composed by count constraints involving only target relations.*

The data posting problem associated with this setting is: given a finite source instance I_S for S, find a finite instance I_T of T such that $\langle I_S, I_T \rangle$ satisfies $\Sigma \cup \Sigma_T$. □

Semantics. The semantic of the Smart Data Posting setting can be done in terms of the standard Data Posting setting. In particular, the standard Data Posting setting corresponding to a given Smart Data Posting setting can be constructed as follows.

Initially, the source and the target schemes as well as the target count constraint can be taken from the Smart Data Posting setting. The domain schema contains the unary domain relation schemes $\mathcal{D}_1$ and $\mathcal{D}_2$. The domain relation $\mathcal{D}_1$ contains values -1 and 1, while $\mathcal{D}_2$ contains values -1, 0 and 1. The set of source to target constraints is empty.

Next, we traduce every non-trivial mapping rule ρ of the form

$$\forall \mathbf{z}[\ \phi_S(\mathbf{z}) \xrightarrow{\mathbf{y}, k, \langle \mathbf{v}, f \rangle} r(\mathbf{x}, \mathbf{v})\ ]$$

into the constructs of the standard setting.

In the following, if $\langle \mathbf{v}, f \rangle$ is omitted, $\mathcal{D}_\rho$ denotes $\mathcal{D}_1$, otherwise $\mathcal{D}_\rho$ denotes $\mathcal{D}_2$. We introduce in the target schema the target relations $A_\rho(\mathbf{X}, \mathbf{Y}, \mathbf{V})$ and $Add_\rho(\mathbf{X}, \mathbf{V}, Flag)$, where $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{V}$ represent vectors of attributes corresponding to the vectors of variables $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{v}$, respectively, while the decision weather to select the pair $\langle \mathbf{x}, \mathbf{v} \rangle$ in the target relation r is stored by the attribute $Flag$: -1 or 0 (not added) and 1 (added).

The set of source to target dependencies is enriched with the following rules:

$$\phi_S(\mathbf{z}) \rightarrow \mathtt{A}_\rho(\mathbf{x}, \mathbf{y}, \mathbf{v})$$
$$\phi_S(\mathbf{z}) \land \mathcal{D}_\rho(\mathtt{flag}) \rightarrow \mathtt{Add}_\rho(\mathbf{x}, \mathbf{v}, \mathtt{flag})$$

where all the variables are universally quantified. The use of the domain relation $\mathcal{D}_\rho$ ensures that only a value among -1, 0 and 1 can be chosen for each $(\mathbf{x}, \mathbf{v})$ in the relation $\mathtt{Add}_\rho$.

Finally, the set of target count constraints is modified as follows. First, we replace every occurrence of $r(\mathbf{z})$ with $\mathtt{Add}_\rho(\mathbf{z}, 1)$. Next, we add the set of constraints that allow us to establish the flag value:

(1) We start by adding *s*upport constraints, that ensure that the value -1 is assigned to the variable $flag$ iff the degree of support of the combination $\langle \mathbf{x}, \mathbf{v} \rangle$ does not reach k. First, we add constraint for values 1 and -1.

$$\mathtt{Add}_\rho(\mathbf{x}, \mathbf{v}, \mathtt{1}) \rightarrow \#(\{\ \mathbf{Y} : \mathtt{A}_\rho(\mathbf{x}, \mathbf{Y}, \mathbf{v})\}) \geq \mathtt{k}$$
$$\mathtt{Add}_\rho(\mathbf{x}, \mathbf{v}, \mathtt{-1}) \rightarrow \#(\{\ \mathbf{Y} : \mathtt{A}_\rho(\mathbf{x}, \mathbf{Y}, \mathbf{v})\}) < \mathtt{k}$$

If $\langle \mathbf{v}, f \rangle$ is not omitted, we also add constraint for value 0:

$$\mathtt{Add}_\rho(\mathbf{x}, \mathbf{v}, \mathtt{0}) \rightarrow \#(\{\ \mathbf{Y} : \mathtt{A}_\rho(\mathbf{x}, \mathbf{Y}, \mathbf{v})\}) \geq \mathtt{k}$$

(2) When f = *unique* the *uniqueness choice constraint* is added:

$$\mathtt{Add}_\rho(\mathbf{x}, \_, \mathtt{flag}) \land \mathtt{flag} \geq \mathtt{0} \rightarrow \#(\{\ \mathbf{V} : \mathtt{Add}_\rho(\mathbf{x}, \mathbf{V}, \mathbf{1})\}) = \mathtt{1}$$

This constraint ensures the exactly one initialization of $\mathbf{v}$ for each $\mathbf{X}$.

(3) When f = *max* we add (i) the *optimization constraint*:

$$\mathtt{Add}_\rho(\mathbf{x}, \mathbf{v_2}, \mathtt{1}), \mathtt{Add}_\rho(\mathbf{x}, \mathbf{v}, \mathtt{0}) \rightarrow \#(\{\mathbf{Y} : \mathtt{A}_\rho(\mathbf{x}, \mathbf{Y}, \mathbf{v_2})\}) \geq \#(\{\mathbf{Y} : \mathtt{A}_\rho(\mathbf{x}, \mathbf{Y}, \mathbf{v})\})$$

(ii) the *choice constraint*, ensuring that at least one initialization of $\mathbf{v}$ for each $\mathbf{X}$ is selected:

$$\mathtt{Add}_\rho(\mathbf{x}, \_, \mathtt{flag}) \land \mathtt{flag} \geq \mathtt{0} \rightarrow \#(\{\ \mathbf{V} : \mathtt{Add}_\rho(\mathbf{x}, \mathbf{V}, \mathbf{1})\}) \geq \mathtt{1}$$

(4) When f = *unique* $\land$ *max* we add the uniqueness choice constraint and the optimization constraint.

4 COMPLEXITY RESULTS

In this section we perform a complexity analysis of the framework.

THEOREM 1. *Given a Smart Data Posting setting (S, T, Σ, Σ_T) and a finite source instance I_S for S, the problem of deciding whether there exists a finite instance I_T of T such that $\langle I_S, I_T \rangle$ satisfies $\Sigma \cup \Sigma_T$ is NP-complete under the data complexity.*

Proof. Membership to NP is obvious: it is sufficient to guess an instance I_T of T and to check whether or not $\langle I_S, I_T\rangle$ satisfies $\Sigma \cup \Sigma_T$. Observe that the size of I_T is polynomially bounded by the input size as no duplicated tuples are allowed in a relation. Furthermore, it is easy to see that checking all constraints on I_T can be easily done in deterministic polynomial time.

To prove NP-hardness we next produce a reduction from the graph 3-coloring, which is well known to be NP-complete. Take any (undirected) graph $G = (N, A)$, where N is the set of nodes and $A \subseteq N \times N$ is the set of arcs. We are also given three colors, say g, r and b.

We define a source scheme S consisting of the relations $nodes_S(N)$, $arcs_S(N_1, N_2)$, and $colors_S(C)$, storing the nodes and arcs of the graph and the admissible (three) colors, respectively. The target database scheme T contains the relations $arc_T(N_1, N_2)$ and $cn_T(N, C)$ describing the arcs of the graph and the node's colors, respectively.

The set of Σ of smart mapping rules is composed by the rules:

$$(1): \ \mathtt{arc_S(n_1, n_2)} \longrightarrow \mathtt{arc_T(n_1, n_2)}$$

$$(2): \ \mathtt{nodes_S(n)} \wedge \mathtt{colors_S(c)} \xrightarrow{\langle \mathtt{c, unique}\rangle} \mathtt{cn_T(n, c)}$$

that simply copy the content $arcs_S$ relations (rule 1) and assign a unique color to every node in a non deterministic way (rule 2).

The set Σ_T is composed by the following count constraint:

$$\begin{aligned}(3): \ \mathtt{arc_T(n_1, n_2), cn_T(n_1, c_1), cn_T(n_2, c_2)} \longrightarrow \\ \#(\{Y : Y = c_1 \wedge Y = c_2\}) = 0\end{aligned}$$

that ensures that the nodes of an arc have different colors.

It turns out that the data posting problem admits a solution if and only if the graph is 3-colorable. □

The Smart Data Posting setting is called *Semi-deterministic* if the *non determinism* in the data posting process *is locally-resolvable*, i.e. if each target relation defined by the mapping rule with uniqueness requirement is not involved in Σ_T.

Theorem 2. *Given a Semi-deterministic Smart Data Posting setting* (S, T, Σ, Σ_T) *and a finite source instance* I_S *for* S*, the problem of deciding whether there exists a finite instance* I_T *of* T *such that* $\langle I_S, I_T\rangle$ *satisfies* $\Sigma \cup \Sigma_T$ *is polynomial under the data complexity.*

Proof. The application of smart mapping rules requires selection, projection, join and aggregation operations. In the case of the uniqueness requirement, one value can be selected arbitrarily. Thus, the number of tuples that can be added to the target instance is polynomial in the size of the relations and the domains that occur in their bodies.

Since each target relation defined by the non-trivial mapping rule is defined by only this rule, its instance is generated following the indication of this rule. The unique case, than the process is non-deterministic regards the uniqueness requirement. Since the target relations defined by the smart mapping rules with uniqueness requirements are not involved in Σ_T, their generated instances will trivially satisfy Σ_T.

Once generated all possible tuples of T, the next step consists in verifying whether the tuples in T satisfy all target constraints. This check is obviously performed in polynomial time. □

Corollary 1. *Given a Semi-deterministic Smart Data Posting setting* $(S, T, \Sigma, \emptyset)$ *and a finite source instance* I_S *for* S*, a finite instance* I_T *of* T *such that* $\langle I_S, I_T\rangle$ *satisfies* Σ *always exists and can be found in polynomial time (under the data complexity).*

Proof. Straightforward from Theorem 2. □

5 CONCLUSION AND FUTURE WORK

In this paper we presented a simplified version of Data Posting framework, based on the use of smart mapping rules. We showed that the data posting problem in the new setting is NP- complete and identify the conditions when this problem becomes polynomial even in the presence of non-deterministic choices.

The proposed approach have been tested in a real scenario within the MISE Project Data Alliance (D-ALL). More in detail, we implemented a prototype that leverages the proposed framework in order to propose users a set of interesting analysis dimensions. Users can validate the proposed dimensions, in that case they are added to the system knowledge base. Our early experiments are quite encouraging and will be deeply refined as a future work.

The simplification of data posting framework proposed in this paper is based on the idea to integrate local selection criteria in the mapping process. The mapping rules, or similar formalism can be also profitable used in different logic-based settings (i.e., P2P Deductive Databases [6, 7], prioritized reasoning in logic programming [5, 21], efficient evaluation of logic programs [14, 18], etc.).

6 ACKNOWLEDGEMENTS

All authors have been supported by MISE Project Data Alliance (D-ALL).

REFERENCES

[1] D. Agrawal et al. Challenges and opportunities with big data. A community white paper developed by leading researchers across the United States. 2012.

[2] M. Arenas, P. Barceló, R. Fagin, and L. Libkin. Locally consistent transformations and query answering in data exchange. In C. Beeri and A. Deutsch, editors, *PODS*, pages 229–240. ACM, 2004.

[3] M. Calautti, S. Greco, C. Molinaro, and I. Trubitsyna. Rewriting-based check of chase termination. In *Proc. of the 9th Alberto Mendelzon International Workshop on Foundations of Data Management, Lima, Peru*, 2015.

[4] M. Calautti, S. Greco, C. Molinaro, and I. Trubitsyna. Exploiting equality generating dependencies in checking chase termination. *PVLDB*, 9(5):396–407, 2016.

[5] L. Caroprese, I. Trubitsyna, and E. Zumpano. A framework for prioritized reasoning based on the choice evaluation. In *Proc. ACM Symposium on Applied Computing (SAC), Seoul, Korea, March 11-15, 2007*, pages 65–70, 2007.

[6] L. Caroprese and E. Zumpano. Aggregates and priorities in P2P data management systems. In *Proc. of 15th International Database Engineering and Applications Symposium (IDEAS 2011), Lisbon, Portugal*, pages 1–7, 2011.

[7] L. Caroprese and E. Zumpano. Computing a deterministic semantics for P2P deductive databases. In *Proc. 21st International Database Engineering & Applications Symposium, IDEAS 2017, Bristol, United Kingdom*, pages 184–191, 2017.

[8] N. Cassavia, E. Masciari, C. Pulice, and D. Saccà. Discovering user behavioral features to enhance information search on big data. *TiiS*, 7(2):7:1–7:33, 2017.

[9] A. Deutsch, A. Nash, and J. B. Remmel. The chase revisited. In *Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada*, pages 149–158, 2008.

[10] R. Fagin, P. G. Kolaitis, and L. Popa. Data Exchange: getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, 2005.

[11] B. Fazzinga, S. Flesca, E. Masciari, and F. Furfaro. Efficient and effective RFID data warehousing. In *International Database Engineering and Applications Symposium (IDEAS 2009), September 16-18, 2009, Cetraro, Calabria, Italy*, pages 251–258, 2009.

[12] N. Fiorentino, S. Greco, C. Molinaro, and I. Trubitsyna. ACID: A system for computing approximate certain query answers over incomplete databases. In *Proc. of SIGMOD Conference 2018, Houston, TX, USA*, pages 1685–1688, 2018.

[13] A. D. Francesco, S. Greco, F. Spezzano, and I. Trubitsyna. Chaset: A tool for checking chase termination. In *Scalable Uncertainty Management - 5th International Conference, SUM 2011, Dayton, OH, USA*, pages 520–524, 2011.

[14] G. Greco, S. Greco, I. Trubitsyna, and E. Zumpano. Optimization of bound disjunctive queries with constraints. *TPLP*, 5(6):713–745, 2005.

[15] S. Greco, C. Molinaro, and F. Spezzano. *Incomplete Data and Data Dependencies in Relational Databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2012.

[16] S. Greco, C. Molinaro, and I. Trubitsyna. Computing approximate query answers over inconsistent knowledge bases. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden.*, pages 1838–1846, 2018.

[17] S. Greco, C. Molinaro, and I. Trubitsyna. Approximation algorithms for querying incomplete databases. *Information Systems (2019)*, https://doi.org/10.1016/j.is.2019.03.010, 2019.

[18] S. Greco, C. Molinaro, I. Trubitsyna, and E. Zumpano. NP datalog: A logic language for expressing search and optimization problems. *TPLP*, 10(2):125–166, 2010.

[19] S. Greco, F. Spezzano, and I. Trubitsyna. Stratification criteria and rewriting techniques for checking chase termination. *PVLDB*, 4(11):1158–1168, 2011.

[20] S. Greco, F. Spezzano, and I. Trubitsyna. Checking chase termination: Cyclicity analysis and rewriting techniques. *IEEE Trans. Knowl. Data Eng.*, 27(3):621–635, 2015.

[21] S. Greco, I. Trubitsyna, and E. Zumpano. On the semantics of logic programs with preferences. *J. Artif. Intell. Res.*, 30:501–523, 2007.

[22] L. Libkin. Incomplete data: what went wrong, and how to fix it. In *Proc. Symposium on Principles of Database Systems (PODS)*, pages 1–13, 2014.

[23] T. Lukasiewicz, E. Malizia, and C. Molinaro. Complexity of approximate query answering under inconsistency in datalog+/-. In *Proc. 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden*, pages 1921–1927, 2018.

[24] G. Manco, E. Masciari, and A. Tagarelli. A framework for adaptive mail classification. In *14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2002), 4-6 November 2002, Washington, DC, USA*, page 387, 2002.

[25] E. Masciari. Trajectory clustering via effective partitioning. In *Flexible Query Answering Systems, 8th International Conference, FQAS 2009, Roskilde, Denmark, October 26-28, 2009. Proceedings*, pages 358–370, 2009.

[26] E. Masciari, G. M. Mazzeo, and C. Zaniolo. Analysing microarray expression data through effective clustering. *Inf. Sci.*, 262:32–45, 2014.

[27] E. Masciari, D. Saccà, and I. Trubitsyna. Simple user assistance by data posting. In *Proc. of the 2nd IEEE International Conference on Artificial Intelligence and Knowledge Engineering, AIKE 2019*, pages 1–8, 2019, to appear.

[28] M. Meier, M. Schmidt, and G. Lausen. On chase termination beyond stratification. *PVLDB*, 2(1):970–981, 2009.

[29] A. Onet. The chase procedure and its applications in data exchange. In *Data Exchange, Integration, and Streams*, pages 1–37. 2013.

[30] D. Saccà and E. Serra. Data Exchange in Datalog Is Mainly a Matter of Choice. In P. Barceló and R. Pichler, editors, *Datalog*, volume 7494 of *Lecture Notes in Computer Science*, pages 153–164. Springer, 2012.

[31] D. Saccà, E. Serra, and A. Guzzo. Count Constraints and the Inverse OLAP Problem: Definition, Complexity and a Step toward Aggregate Data Exchange. In T. Lukasiewicz and A. Sali, editors, *FoIKS*, volume 7153 of *Lecture Notes in Computer Science*, pages 352–369. Springer, 2012.

[32] Special report. Big data. *Nature*, Sept. 2008.

[33] Special report. Data, data everywhere. *The Economist*, Feb. 2010.

Anonymously forecasting the number and nature of firefighting operations

Jean-François Couchot, Christophe Guyeux
Femto-ST Institute, UMR 6174 CNRS,
University of Bourgogne-Franche-Comte
Belfort, France
jean-francois.couchot@univ-fcomte.fr,christophe.guyeux@univ-fcomte.fr

Guillaume Royer
Service Départemental d'Incendie et de Secours du Doubs
(SDIS 25)
Besançcon, France
guillaume.royer@sdis25.fr

ABSTRACT

Predicting the number and the type of operations by civil protection services is essential, both to optimize on-call firefighters in size and competence, to pre-position material and human resources... To accomplish this task, it is required to possess skills in artificial intelligence, which are not usually found in a medium-sized fire department. However, such a request may be mandated, for example from specialized companies or research laboratories. This mandate requires the transmission of potentially sensitive information relating to interventions which is not intended to be publicly available. The purpose of this article is to show that a machine learning tool can be deployed and provide accurate results, using a learning process based on anonymized data. Learning on real but anonymized data will be performed using extreme gradient boosting, and the performance of each anonymization will be compared on the number and of interventions per day, and their type.

CCS CONCEPTS

• **Information systems** → *Data stream mining*; • **Security and privacy** → *Usability in security and privacy*; • **Computing methodologies** → *Spatial and physical reasoning*; *Neural networks.*

KEYWORDS

Data Privacy, Data anonymity

ACM Reference Format:
Jean-François Couchot, Christophe Guyeux and Guillaume Royer. 2019. Anonymously forecasting the number and nature of firefighting operations. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3331076.3331085

1 INTRODUCTION

For various economic and societal reasons, such as the aging of the population, the closure of small rural hospitals, or the disengagement of the private sector (ambulance drivers) for acts that are not economically interesting, French fire brigades are facing a constant increase in the number of interventions. However, due to the economic crisis and the state debt, the resources allocated to public services in general, and to the fire brigade in particular, are not increasing on their side. The latter must therefore find original solutions to meet growing demand in constant resource. One solution for the future is to optimize the use of their human and material resources, by pre-positioning vehicles and adapting the size of the guards according to the number, type and location of intervention that an artificial intelligence algorithm could predict.

This solution requires, on the one hand, a database of past interventions that is sufficiently rich and consistent, and on the other hand, know-how in a constantly evolving scientific discipline. This knowledge base is naturally present within the departmental fire and rescue service (SDIS), which collects, for legal and statistical purposes, many data related to each of their interventions. This database contains information on the dates, places and types of interventions, as well as on the interveners and victims. However, if the SDIS has this basis of knowledge useful in the learning phase of an artificial intelligence algorithm, it has neither the know-how nor the human resources to implement such an algorithm.

Indeed, such a realization implies the recovery of explanatory variables by scripts automatically retrieving internet information on past meteorology, ephemerides, epidemiological data, etc. Selecting models from among the various machine learning methods based on decision trees or artificial neurons, as well as feature selection to reduce model complexity, requires time and up-to-date knowledge of machine and deep learning techniques. Similarly, finding good values for algorithm hyperparameters, or proposing resource optimizations based on predictions made, requires the work of computer researchers specialists in artificial intelligence, high performance computing, and optimization.

If the basis of knowledge, with the personal data it contains, is legally protected as long as it remains within the SDIS, its complete transmission to another institution, even if it remains public, is problematic, at least legally. Therefore, the data must be de-identified and then processed by academics, with no intention of public disclosure. However, if anonymization of the data is mandatory to allow such transmission from SDIS to the university, this anonymization should not make the data unusable for any type of prediction. In other words, a fair compromise should be found between the protection of private information contained in the database and the amount of preserved information useful for machine learning algorithms. In fact, the question of whether such a compromise exists and can be found is worth asking.

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331085

The objective of this article is to present a concrete case of fine optimization by state-of-the-art techniques, making it possible to guarantee both a sufficiently high privacy given the context (private exchange between fire brigades and academics), while allowing better predictions than what could be obtained with traditional statistical tools. It is therefore a proof of concept on a concrete case study from the SDIS 25 (firemen from Doubs department in France), showing that a fair compromise is possible, allowing a future optimization of firefighters' resources without paying for it by potential leaks in privacy.

The rest of this article is structured as follows. The case study is presented in the next section, which contains a description of the data under consideration. Section 3 focuses on the problem of de-identification with an overview of most important methods that have been applied on this case study. The database that has been anonymized is then used to learn and predict firemen interventions in Section 4. This article ends by a conclusion section, in which the contribution is summarized and intended future work is outlined.

2 DATA PRESENTATION

The data we have to conduct the forecasts are classified by year between 2012 and 2017. Each intervention of the fire fighters of the fire brigade of the Doubs department (a French county of 500,000 inhabitants) is recorded in a file in the form of a line. The attributes of this file are shown in the Table 1 and described as follows:

ID	Station	Reason	Commune	SDate	Age	Gender	SAD	Type	Destination	Doctor	Condition	Location
0	Belfort South	Malaise	Belfort	2018/01/31 08:35	45	Male	No CRA	Other	Belfort Hospital	No	Severe Injury	(47.616, 6.857)

Table 1: Attributes of fire brigade operations data

- *ID* is the ID intervention, which is used in supplementary files;
- *Station* is the fire station name;
- *Reason* is the initial reason for the firefighters' intervention;
- *Commune* is the name of the municipality where the operation took place;
- *SDate* is the starting date of the intervention
- *Age Gender* and *Type* is the age, the gender of the victim, and whether it is a fireman or not;
- *SAD* indicates whether a Semi-Automatic Defibrillator has been used;
- *Destination* gives the subsequent destination, *i.e.* the place where the firefighters transported the victim later;
- *Doctor* specifies whether a doctor was present at the victim's location;
- *Condition* states the victim's condition at the end of the operation;
- *Location* gives the precise location (latitude, longitude) of the intervention.

The Table 2 gives the number of interventions by firefighters per year. As can be seen in this table and as stated in the introduction, the number of firefighters' operations is constantly increasing.

Year	Number of operations
2012	22,960
2013	24,562
2014	26,026
2015	27,750
2016	28,880
2017	31,715
Total	161,813

Table 2: Number of interventions by firefighters per year

3 DE-IDENTIFICATION PROBLEMS

This section shows how the fire brigade data were de-identified in order to first predict the number of interventions (Sec. 3.1), then to give the kind of intervention (Sec.3.2).

3.1 Number of interventions per fire station by time slot

The objective of this first part is to have firefighters in each center always in adequacy with the interventions to be carried out by the center's personnel. To know the number of firefighters present and/or available in each center, it is necessary to have an idea of the number of interventions per year, per month, per week, per day, per 3-hour block in each of these centers. The objective is to publish data in the form of tuples (*SDate*, *Station*, *#Operations*) where *SDate* is the time interval (with variable amplitude as discussed before), *Station* is the station name, or a generalization. Finally, *#Operations* represents the number of actions performed by the fire fighters of the *Station* unit(s) during the time interval *SDate*.

In small rural centers, where the number of interventions is naturally low, it can happen that the hourly amplitude of the study is too low compared to the number of interventions carried out by the fire brigade of the latter and therefore not significant enough to be generalized. It is therefore natural to think of grouping these stations together at the level of the urban community to obtain events that are sufficiently representative in number.

The first question here is: is the number of interventions a sensitive attribute? Clearly yes. This gives importance to a fact. The movement of the Fire Brigade would not take place if the situation had not been critical. For example, if it is known, on the one hand, that a person was sick in a small village and equipped with a centre and that this centre performed an intervention during this period (whereas it almost never does), then there is a high probability that the fire brigade intervened for this person and therefore that the illness worsened.

Two anonymization approaches are used here as direct applications of existing methods. The first is k-anonymity [8] and the second is differential confidentiality [2].

3.1.1 A k-anonymous de-identified dataset. In order for the article to be self-sufficient, we recall here the definition of the k-anonymity requirement.

Definition 3.1 (k-anonymity requirement[8]). Each release of data must be such that every combination of values of quasi-identifiers can be indistinctly matched to at least k individuals.

In other words, for a given dataset with at least k equivalent records, the probability of re-identifying an individual, for any known given attack A is less than $1/k$.

Thus, only triplets (*SDate*, *Station*, *#Operations*) such as the number of *Operations* is greater or equal to k will be provided for further analysis. The others, (*i.e.*, when *#Operations* $< k$) will not be used in the further prediction step (since they are removed from the dataset), reducing thus the approach accuracy. This raises the question of choosing a value for the k parameter: a high value decreases the overall probability of re-identification but results in a loss in the data's accuracy. The chosen value k has to ensure an acceptable risk of re-identification for any kind of attack A, *i.e.*, $P(\textit{re-identification}|A) \leq \frac{1}{k-1}$ is lower than a given value.

$$P(\textit{re-identification}|A) = \frac{P(\textit{re-identification}, A)}{P(A)} \quad (1)$$

has to be evaluated for each kind of attack A, namely deliberate attempt at re-identification, acquaintance (*i.e*, inadvertent attempt), or breach. For each kind of attack A, the following probability $P(\textit{re-identification}, A)$ must be lower than a commonly acceptable threshold T. Quoting [5], since the dataset will be distributed to researchers only, the average risk threshold T is set to 0.1.

In our context, researchers belong to an academic institution with a confidentiality data agreement, without any particular intent to re-identify records. It is recognized in such a case that $P(\textit{Deliberate Attempt}) \leq 0.4$. The third attack, (breach) can take place if the university loses the dataset. According to [4], it results that $P(\textit{Breach}) = 0.27$. We are then left to evaluate $P(\textit{Acquaintance})$.

The whole dataset is composed of less that 162,000 operations in the Doubs department (composed of 500,000 inhabitants) which may concern the same individual. The probability of an individual not to be in this dataset is about 1-162000/500000=0.676. Since the average estimated number of well-known contacts is 150, the probability that none of them are in the dataset is approximately equal to 0.676^{150}, which is very close to 0. In this context, the probability of acquaintance is thus equal to 1, *i.e.*, $P(\textit{Acquaintance}) = 1$.

The higher the value of $P(A)$, the smaller $P(\textit{re-identification}|A)$ and the more de-identification is required on the data set. One thus have to ensure that $P(\textit{re-identification}|A) \leq \frac{0.1}{1}$, *i.e.* $k = 11$.

Attribute	Generalization Hierarchy
SDate	date-hh:mm:ss → 3-hours → day → week → month
Station	station Name → urban community → county

Table 3: Generalization hierarchy for number of interventions per fire station by time slot

A generalization approach can be applied on both attributes *SDate* and *Station* and is represented on Table 3. It is a list of simplifications which can be applied to attribute values, ordered from the smallest intervals to most general ones. Counting the number of operations in an urban community rather than a fire station aims to reduce the number of deletion in the data set to allow for better learning and prediction: there are fewer cases in an urban community than in a fire station where the number of operations per time interval will be less than k. Of course, the results of the predictions will be given at the level of these communities, but many firefighters live in the metropolitan communities and can move to another fire station if needed. Table 4 gives results of 11-anonymity dataset with respect to the generalization parameters. In each cell, the first number gives the rate of suppressed records whereas the latter is the entropy value [5] expressed as a percentage (compared to the maximum possible entropy for the data set). It can be deduced that the generalization of the starting date to the day, and the fire station to the urban community gives acceptable results both in terms of records loss and entropy.

3.1.2 A ε differential private dataset. Differential privacy [2, 3] is property of anonymization technique that minimizes the privacy impact on individuals whose information is in the database. From a probabilistic point of view, it is not possible for an attacker to identify sensitive data about an individual if his/her information were removed from the dataset. Practically, it may be implemented as noise addition to query results.

Let f be the function that associates to each fire brigade its number of interventions at a given time. If an operation by firemen of this station is deleted, the impact is exactly 1 and the sensitivity of f, usually denoted as Δf, is thus equal to 1. It has been proven that a mechanism that returns $f(x) + y$ where y is the added noise that follows a Laplacian distribution $(0, \frac{\Delta f}{\varepsilon})$ is ε-differential private. A high value of ε leads to small value noise and induces thus a low guarantee of privacy. On the opposite, a small one provides a high probabilistic guarantee against attacks. We are then left to assign a value for the ε parameter with the goal of hiding any individual's presence in the dataset.

According to [6], the value of ε should be bounded by

$$\varepsilon \leq \frac{\Delta f}{\Delta v} \ln \frac{(n-1)\rho}{1-\rho}, \quad (2)$$

where n is the number of lines of our dataset (*i.e.*, at least $n = 22,960$), Δv is the longest distance between two datasets where a line has been removed each time (*i.e.*, $\Delta v = 2$), and ρ is the probability of being identified as present in the database. To be coherent with Sec. 3.1.1, ρ is set with 0.1. In such a case, ε should be lower than 3.92. The value $\varepsilon = 1$ has been retained here.

3.2 Nature of firefighters' interventions by time slot

To optimize the material and human resources present in each station or urban community, it would be interesting to predict the types of interventions by time interval (year/month/week/ week/day/3-hour block) in each area of interest (station, urban community, department).

For a particular time block of a given amplitude, the types of tasks executed (the reason attribute) are extracted from the data set. The cardinal of this set (in which the equal types are deleted) is naturally lower than the number of interventions found in the Section 3.1. The nature of these interventions is clearly a sensitive data.

	SDate	3-hours	**Day**	Week	Month	Year
Fire station	99.8/0.0	99.6/23.1	64.3/70.6	27.6/60.9	5.2/75.4	0.2/99.8
Urban Community	99.8/0.0	96.9/23.1	**32.7/32.8**	7.8/61	0.7/75.5	0.0/99.9
County	99.8/0.19	38.0/38.9	0/42.1	0/61.1	0/75.6	0/100

Table 4: Number of interventions : anonymization by generalization and 11-anonymity

There are ≈400 different reasons in the database for firefighters to be involved, some of them overlapping or are very similar to each other. During each intervention, the reason for departure is indicated at the beginning of the intervention, *i.e.* often in an emergency context, leading to a certain number of errors. The finer the granularity, the more errors there are in an emergency situation. To improve data quality, the reasons for firefighters to leave are therefore regrouped into 7 classes that ares *personal assistance*, *road rescue*, *another accident*, *fire or explosion*, *various operations*, *preventive operations*, *other reasons*. This is like applying a low-frequency filter. Once again, it is a question of finding the right compromise between the usefulness of the data and their quality. In all of the following, we only considered data resulting from the grouping in accordance with this filter.

3.2.1 Recursive (c, l)-diversity. Publishing the types of interventions is critical because if they are not varied enough, then this information can be misused and led to a positive or a negative disclosure. For example, if all the outings that took place on a given date involved heart ailments and if we know that a person was rescued by firemen on that day, we deduce that they had a heart attack. This is the problem identified by Machanavajjhala et al. and named l-diversity [7].

Intuitively, a group of records (bloc, equivalent class) is said to be l-diverse if there are at least l "well-represented" values for the sensitive attributes (which may be a single sensitive attribute, a pair of sensitive attributes, …). The dataset is said to be l-diverse if each group of records is l-diverse. The notion of "well-represented" is intentionally ambiguous. The fact that l separates values is not sufficient for this definition. A potential refinement could be that the current values are distributed according to a law approaching uniform distribution. We then find the notion of Entropy l-diversity. However, this constraint is often overly restrictive.

We prefer to take a less restrictive refinement that stipulates that the ratio between the most represented value and the sum of the least $m - l + 1$ represented ones is less than a constant c provided by the user. This definition is known as recursive (c, l)-diversity [7] and is recalled here.

Definition 3.2 (Recursive (c, l)-Diversity). In a given q^*-block, let r_i denote the number of times the i^{th} most frequent sensitive value appears in that q^*-block. Given a constant c, the q^*-block satisfies recursive (c, l)-diversity if $r_1 < c(r_l + r_{l+1} + \cdots + r_m)$. A table T satisfies recursive (c, l)-diversity if every q^*-block satisfies recursive (c, l)-diversity.

The higher the c number is, the more frequently this property is established.

From the experiments in Section 3.1.1 concerning 11-anonymity, we focused on the generalization of fire stations at the level of the agglomeration community and the date of intervention at the level of the day. For this given generalization, we varied l and c both in $\{2, 3, 4, 5, 6\}$. Results are summarized in Table 5 where for each pair (l, c), the rate of suppressed records is given.

In this table and not surprisingly, the number of deleted records is decreasing with respect to c, but increasing w.r.t. of l. Results of de-identification satisfying recursive $(5, 2)$-diversity has been thus retained because it is the only one that does not delete all the data.

l/c	2	3	4	**5**	6
2	93.5	85.0	76.0	**68.0**	61.3
3	99.9	99.7	99.5	99.2	98.6
4	100	100	100	100	100
5	100	100	100	100	100

Table 5: Reasons of interventions: rate of suppressed records with anonymization by generalization, 11-anonymity and recursive (c, l)-diversity

.

3.2.2 Differentially Private Histogram of Operations. In [9], Xu et al. show how to publish a differentially private compliant histogram which outputs the distribution of a random variable, such as the number of operations with respect ot the attribute *Reason* of intervention.

The approach is twofold. In the former, for a given time slot (a whole day, e.g.), a histogram is constructed representing the number of interventions performed during this period and whose values are grouped according to the reasons for the intervention of the fire brigade. In the latter, a noise is added with unit-length bins, using the Laplace Mechanism. The resulting histogram is thus published for analyzes. The clustering of reasons for departures into 7 classes (as presented in the beginning of this section) was particularly guided by this step. Indeed, without this grouping, a histogram with potentially 400 bars can be constructed (there are approximately 400 different reasons of intervention, as presented at the beginning of this section). Even with the addition of very low random noise, some of the reasons may appear when they are not at all correlated with an event. By grouping the reasons into 7 classes, the granularity is certainly less, but the added noise is still meaningful.

As in Sec. 3.1.2, ε should be chosen to respect privacy concern. Even if the request executed here on the database is different than the one given in this section, all the variable values of Equation (2) are the same leading to a bound for ε which is 3.92. In what follows, ε is thus set again with $\varepsilon = 1$.

4 MACHINE LEARNING PREDICTIONS

4.1 General presentation

The objective of this section is now to evaluate whether it is possible to make predictions about the activity of firefighters from anonymized files. We will focus on the number of interventions per unit of time, the type of intervention, and the solicitation per centre. In each case, predictions based on anonymized data will be compared to those based on raw data. More specifically, we will look at whether, from the anonymized data of 2013-2017, we can find out what happened in 2012, as described in the anonymized file of 2012. This predictive ability will be compared to the score obtained by predicting the year 2012 (not anonymized) from the learning on the raw data for 2013-2017. Finally, the 2012 prediction based on the anonymized 2013-2017 data will be compared to the de-anonymized 2012. Note that we have chosen to predict 2012 from 2013-2017, and not 2017 from 2012-2016, because the year 2017 saw its number of interventions explode due to a disengagement of the private sector (ambulance drivers) artificially inflating firefighters' interventions, and this for reasons that are difficult to predict because they are no longer linked to human activity: instead of predicting the future, we are reconstructing a potentially unstored past.

In order to achieve this supervised learning, we had to recover a collection of explanatory variables that could potentially explain the number, type and location of interventions. We have assumed that these interventions are directly related to human activity (for example, there is less intervention at night, because people sleep), which itself changes according to the time of year (holidays, seasons...), the weather, etc. These explanatory variables, for each hour of the period under consideration, are publicly available on the Internet, and have enabled us to recover with some precision the 2012 interventions from those of 2013-2017.

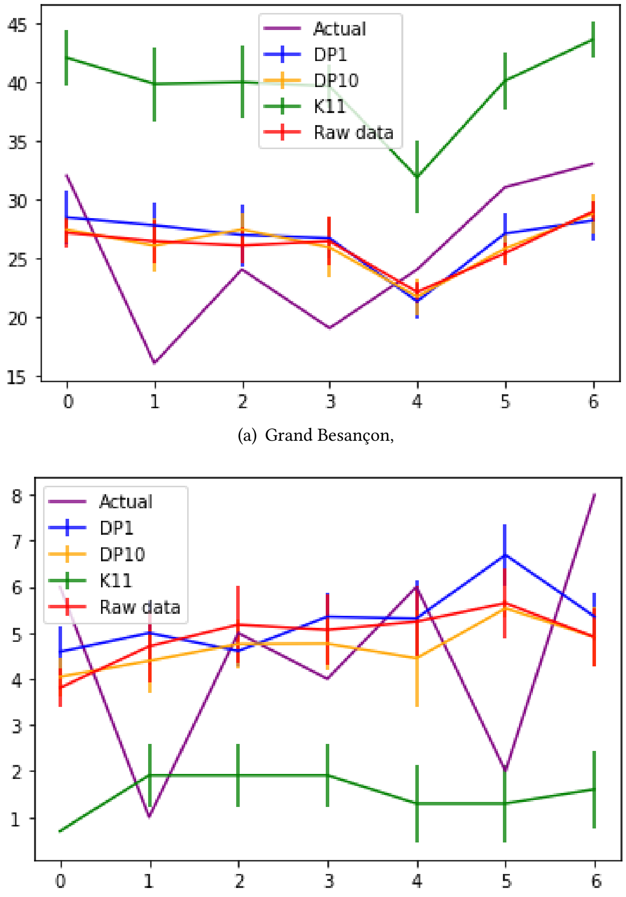
In detail, the following numerical variables were recovered from the MétéoFrance site for the three weather stations closest to the Doubs (Nancy, Dijon and Basel): wind direction, humidity, dew point, precipitation during the previous hour, and during the last 3 hours; pressure, and its variation lods over the last 3 hours, temperature, wind speed, and finally visibility. At the calendar level, we have added the year, month, day in the week (Monday...), in the month (1.2, ..., 31) and in the year, in order to identify days different from the normal (national holiday, Christmas...). Epidemiological data have also been added on the incidence of influenza, chickenpox and diarrhoea over the past week, collected from the Sentinel network. Finally, since the Doubs department is rich in mountains, forests and rivers, in a temperate region, we occasionally have heavy rainfall leading to sudden variations in the height of the rivers. The latter lead to floods, requiring assistance to people. Also, the heights of six rivers have been added, with their variations over the past hour.

In the following, we will present the prediction results from approaches that can be obtained using the explanatory variables on original data, and finally what is found using the anonymized version of the data. Finally, it should be noted that the machine learning algorithm used here is the extreme gradient boosting (XGBoost), with the default values as hyperparameters [1]. For each set of prediction attempts, 5 experiments have been done with distinct seeds for random initialization. Each curve represents the curve of the means and the standard deviation is always displayed with vertical bars.

4.2 Predicting the number of interventions

In this section, the objective is to predict the number of interventions per fire station by time slot. The de-identification has shown that an acceptable trade-of between the number of suppressed data, the entropy and the duration of the study is obtained by merging data inside Urban Communities and for a duration equal to the day (see 4).

Let us first recall that ensuring 11-anonymity had a cost: a number of lines have been deleted. More precisely, 32.7% of the interventions were removed, and therefore a factor multiplying the number of predicted interventions by 1/(1-32.7/100) = 1.486 will be considered. For this method, this corresponds to an adjustment achieved by a systematic increase in forecasts of about 49%.



(a) Grand Besançon,

(b) Grand Pontarlier,

Figure 1: Predictions Week # 2, 2012

Three anonymisation methods have been applied, namely 11-anonymity and Differential Privacy with $\varepsilon = 1$ and $\varepsilon = 10$. Figure 1 and all subsequent ones compare the results of predictions from the original data, anonymized data using these three methods, and the mean (which is the simplest prediction). Each time, these predictions are given for the week between January 8 and 14, chosen as an arbitrary example. All these results focus on two agglomeration communities, namely Grand Besançon and Grand Pontarlier. These two agglomerations were chosen because their demographic characteristics are significantly diverse. The former has about 200,000 inhabitants living in 68 municipalities, mainly urban over a surface area of 528.6 km2, representing approximately 379 inhabitants/km2. The latter is composed of approximately 27,000 inhabitants who live in 10 municipalities, mainly rural, over an area of 154 km2, that is, approximately 175 inhabitants/km2.

In this figure and in all the following, Actual (in purple) always denotes what really happens during this week. Concerning predictions, K11 (in green) is the average curve of predictions through 11-anonymity concept. DP1 and DP10 are the mean curves after Differential Privacy based anonymization with $\varepsilon = 1$ and $\varepsilon = 10$ respectively. Red curves represent data forecast from original row data (*i.e.*, non de-identified data).

Let us explain results obtained for the agglomeration of Grand Besançon (Figure 1(a)). First of all, the average number of interventions for this agglomeration is 28.1 with a standard deviation of 7.7. The Mean Absolute Error (MAE) when considering this average as a prediction of reality is 6.6. Any prediction based on intelligence must reduce this error.

Note that all predictions based on Differential privacy and on row data are consistent: the standard deviation for each prediction is about 6.6. Here, the prediction with data anonymized with 11-anonymity is over-estimated: as already announced, this method has lead to a suppression of 32.7% of data indeed. But only a few of removed data concerns Grand Besançon and the forecast accuracy for this urban community was thus sufficient enough. However all the predictions with this generalization based anonymization method have been increased by 49% leading here to a over-estimation. In this agglomeration, even if the predictions are not extremely accurate, we can see that they follow the same trend as reality: a relative decreasing until the middle of the week with a more or less rapid ascendancy thereafter. The mean average errors w.r.t the chosen anonymisation method are reported in Table 6. It can be seen in the latter that the predictions on data anonymized by Differential Privacy have the same level of accuracy as those from the original data.

The results are much less homogeneous for the Pontarlier urban community (Figure 1(b)). In this case, the predictions from the data anonymized by 11-anonymity are far below reality and other predictions. This is explained by the fact that in this urban community, the average number of interventions for this week is 4.7 with a standard deviation of 2.6. Many data concerning this agglomeration community are thus deleted by the 11-anonymity method. The average number of interventions using this latter anonymization method is indeed 1.7, after the adjustment of the data by 1.5. However, this result is far below reality. The other approaches based on Differential Privacy anonymization methods give forecast which are in the consistent order of magnitude. Regarding the MAE of predictions (Table 6) and as in the other agglomeration community, predictions are as accurate when using data anonymized by Differential Privacy as when embedding raw data.

	Grand Besançon	Grand Pontarlier
Average number of intervention	28.1	4.7
Mean	6.6	2.0
11-anonymity	16.0	3.6
DP ($\varepsilon = 1$)	5.6	1.9
DP ($\varepsilon = 10$)	5.5	1.9
Raw data	5.7	1.9

Table 6: Mean Average Error with respect to anonymization method

Another positive point is that we find, in general, the same relative importance of each explanatory variable: the same causes explaining the number of interventions are highlighted (causality is not confused): the five most important features as provided by the plot_importance function (namely, the year, wind direction, day in the year, humidity, and water level of the Doubs River) are the same, but not in the same order. Let us also note to relativize that, on anonymized data, we obtain predictions that are not totally meaningless (compared to the average), while:

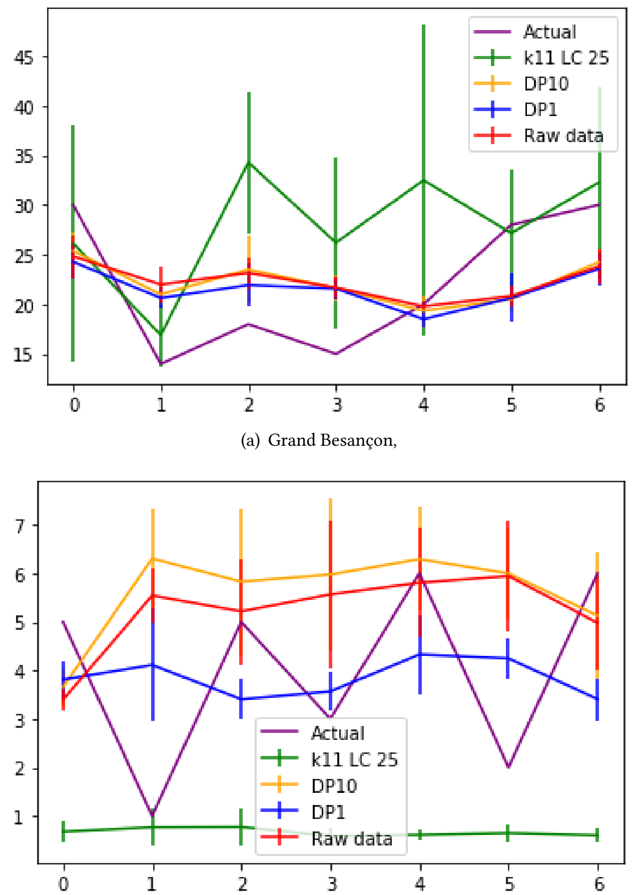
- no model selection (choice of the machine learning algorithm) has been performed;
- no preliminary step was taken to select explanatory variables;
- no attempt was made to optimize the many hyperparameters of the XGBoost.

4.3 Predicting the nature of interventions

In this set of experiments, two anonymisation methods have been applied. The former is 11-anonymity combined with recursive $(5, 2)$-diversity and the latter is histogram of operations compliant with Differential Privacy (with $\varepsilon = 1$ and $\varepsilon = 10$). For the same reasons as above, this study focuses on the two agglomeration communities, Grand Besançon and Grand Pontarlier. This article focuses only on two types of intervention, namely *personal assistance* and *road rescue*. Personal assistance is indeed very frequent and can usually be managed by several services: the SAMU, private ambulances and fire brigades. In contrast, road accidents are more infrequent (and predictable with probably less accuracy), but are systematically handled by firefighters. Results of predictions are shown in Figures 2 and 3. The former deals with personal assistance whereas the latter focuses on road accidents. As in the previous section and for the same reasons, this figures focus on two agglomeration communities, namely Grand Besançon and Grand Pontarlier and the color codes are the same than in previous section.

Let us first focus on personal assistance. As in the previous section, the number of interventions realized for this reason of departure is overestimated when anonymization is achieved by 11-anonymity and recursive (5,2)-diversity when the urban community is Grand Besançon and underestimated otherwise. It happens that data containing this reason may be deleted by this method.

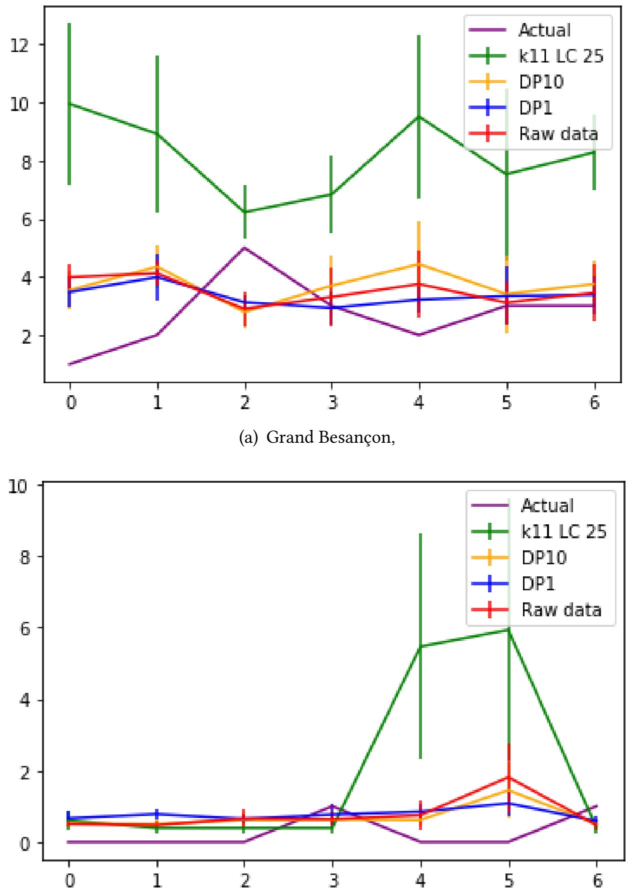
For a medium-sized urban community such as Besançon, the trend is observed also on anonymized data, even if it is slightly overestimated. This is explained by the fact that the number of



(a) Grand Besançon,

(b) Grand Pontarlier,

Figure 2: Personal assistance, Predictions Week # 2, 2012



(a) Grand Besançon,

(b) Grand Pontarlier,

Figure 3: Road intervention, Predictions Week # 2, 2012

interventions for this reason of intervention has increased steadily over the years (between 2012 and 2017) and that 2012 is therefore the year in which these exits have been the least numerous. For the small urban community of Pontarlier, the trend is also found even on data anonymized by the method combining histograms and differential confidentiality. For ε equal to 1 (which guarantees acceptable safety), predictions close to the mean are found.

Road accidents are quite uncommon and therefore more difficult to predict, especially in small rural communities. In this context, it seems even less relevant to apply the 11-anonymity and recursive (5,2)-diversity based anonymization method to make subsequent predictions. This is confirmed by the curves of the figures 3(b) and 3(a). The standard deviation for this anonymization method are indeed very large, and forecast a very far from reality.

The noise added by the method combining histograms and differential confidentiality is sufficiently limited even when ε=1. Indeed, the general trend is found with almost as much precision on data anonymized by such an approach as on raw data, *i.e.*, on non-anonymized data. This trend is numerically validated by the values given in Table 7, which summarizes the mean absolute errors by agglomeration community, by type of intervention and according to the chosen anonymization method. As in the previous table on prediction errors concerning the number of interventions, it can be seen here that the histogram method with differential privacy allows to obtain predictions as precise as those obtained on raw data.

	Grand Besançon		Grand Pontarlier	
	Personal assistance	Road accident	Personal assistance	Road accident
Average number of intervention	24.1	3.4	4.0	0.6
Mean	5.6	2.0	1.7	0.8
11-anon. + recursive (5,2) diversity	7.6	3.9	3.3	0.9
DP (ε = 1)	4.5	2.0	1.7	0.8
DP (ε = 10)	4.6	2.0	1.7	0.9
Raw data	4.5	2.0	1.6	0.8

Table 7: Mean Average Error with respect to anonymization method

5 CONCLUSION

"'Can we predict and with which accuracy the number (1) and nature (2) of firefighters' interventions in a geographical area while respecting the privacy of the victims they rescued?" This article is a positive answer. In both the quantitative (question (1)) and qualitative (question (2)) domains, this article shows that differential confidentiality based approaches provide more accurate results than generalization and suppression ones. It is possible to use privacy-respecting (*i.e.*, properly anonymized) data to guess an accurate behavior.

It should be noted that the variable ϵ was deliberately set to 1 to ensure a high level of privacy. By increasing this value (up to the calculated threshold 3.9), the obtained results would have been even more accurate.

The prospects for this work are numerous. We will first study the possibility of predicting the places of intervention, knowing that this attribute is very critical, because it almost allows the victim to be identified.

This study has been supported by the EIPHI Graduate School (contract "ANR-17-EURE-0002"), by the Interreg RESponSE project, and by the SDIS25 firemen brigade.

REFERENCES

[1] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 785–794. https://doi.org/10.1145/2939672.2939785

[2] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings (Lecture Notes in Computer Science)*, Shai Halevi and Tal Rabin (Eds.), Vol. 3876. Springer, 265–284. https://doi.org/10.1007/11681878_14

[3] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[4] Khaled El Emam and Luk Arbuckle. 2013. *Anonymizing health data: case studies and methods to get you started.* " O'Reilly Media, Inc.".

[5] Khaled El Emam, Fida Kamal Dankar, Romeo Issa, Elizabeth Jonker, Daniel Amyot, Elise Cogo, Jean-Pierre Corriveau, Mark Walker, Sadrul Chowdhury, Regis Vaillancourt, et al. 2009. A globally optimal k-anonymity method for the de-identification of health data. *Journal of the American Medical Informatics Association* 16, 5 (2009), 670–682.

[6] Jaewoo Lee and Chris Clifton. 2011. How Much Is Enough? Choosing ϵ for Differential Privacy. In *Information Security, 14th International Conference, ISC 2011, Xi'an, China, October 26-29, 2011. Proceedings (Lecture Notes in Computer Science)*, Xuejia Lai, Jianying Zhou, and Hui Li (Eds.), Vol. 7001. Springer, 325–340. https://doi.org/10.1007/978-3-642-24861-0_22

[7] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. *L*-diversity: Privacy beyond *k*-anonymity. *TKDD* 1, 1 (2007), 3. https://doi.org/10.1145/1217299.1217302

[8] Pierangela Samarati and Latanya Sweeney. 1998. *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.* Technical Report. technical report, SRI International.

[9] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. 2013. Differentially private histogram publication. *The VLDB Journal—The International Journal on Very Large Data Bases* 22, 6 (2013), 797–822.

Formal Specification and Verification of User-centric Privacy Policies for Ubiquitous Systems

Rezvan Joshaghani
rezvanjoshaghani@u.boisestate.edu
Boise State University
Boise, Idaho

Stacy Black
stacyblack@u.boisestate.edu
Boise State University
Boise, Idaho

Elena Sherman
elenasherman@boisestate.edu
Boise State University
Boise, Idaho

Hoda Mehrpouyan
hodamehrpouyan@boisestate.edu
Boise State University
Boise, Idaho

ABSTRACT

As our society has become more information oriented, each individual is expressed, defined, and impacted by information and information technology. While valuable, the current state-of-the-art mostly are designed to protect the enterprise/ organizational privacy requirements and leave the main actor, i.e., the user, uninvolved or with the limited ability to have control over his/her information sharing practices. In order to overcome these limitations, algorithms and tools that provide a user-centric privacy management system to individuals with different privacy concerns are required to take into the consideration the dynamic nature of privacy policies which are constantly changing based on the information sharing context and environmental variables. This paper extends the concept of contextual integrity to provide mathematical models and algorithms that enables the creations and management of privacy norms for individual users. The extension includes the augmentation of environmental variables, i.e. time, date, etc. as part of the privacy norms, while introducing an abstraction and a partial relation over information attributes. Further, a formal verification technique is proposed to ensure privacy norms are enforced for each information sharing action.

CCS CONCEPTS

• **Security and privacy** → **Logic and verification**; • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Privacy, Formal Methods, User-Centric Policies

ACM Reference Format:
Rezvan Joshaghani, Stacy Black, Elena Sherman, and Hoda Mehrpouyan. 2019. Formal Specification and Verification of User-centric Privacy Policies for Ubiquitous Systems. In *23rd International Database Engineering Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331076.3331105

1 INTRODUCTION

A Privacy Bill of Rights was endorsed by the White House in 2012, a response to an increasingly loud objection of citizens on the lack of privacy and fair information practices guidelines [20]. The predicament was not only recognized by the US government, but has also been investigated and studied at the international stage and has resulted in reports such as "Rethinking personal data: Strengthening trust" by the World Economic Forum (WEF) [40] and "Recommendations for businesses and policymakers" by the Federal Trade Commission (FTC) [12]. Despite all these efforts, ubiquitous online monitoring of users' activities [29] and scandalous data breaches, i.e. Facebook and Cambridge Analytica, continue to haunt Online Social Network (OSN) users [2, 11]. These privacy breaches are often due to a lack of regulatory standardization. Hence, the onus is on the user to take control of: what types of information should be shared with whom and when. However, controlling and managing the information sharing parameters could be a cumbersome and difficult process [15, 21, 44]. Therefore, ample tools and algorithms should be developed and provided to users so they are able to define and enforce their own customized, unambiguous privacy policies and have control over how their information is shared. The state-of-the-art research on privacy management mostly consist of: access control languages [4, 33, 39], different privacy settings in applications, and formal privacy policies [5, 10, 14, 22, 36]. While valuable, the previous works are mostly based on enterprise/organizational privacy management and leave the main actor, i.e., the user, uninvolved or with limited ability to control the information sharing parameter. In addition, existing privacy regulations like HIPAA or a corporation's privacy policies are domain-specific and static with a little or no change over time. On the other hand, the user's privacy policies are dynamic and changing based on many factors, i.e. context, environment, relationship status, etc. In addition to dynamicity, the privacy framework should provide the user with the ability to adapt the policies to their own personal needs, since the definition of privacy varies from person to person based on their personality, cultural background, etc [30].

In order to move towards a more practical solution, this paper proposes a framework to build a user-centric privacy management

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331105

system. We focus on developing the main core of this framework, which is the *privacy formalization and verification engine* that allows for the guided and flexible specification of users' privacy intentions. The formalization and verification engine performs formal reasoning about the user's privacy rules to detect privacy violations. Further, the proposed approach ensures that the defined privacy policy is unambiguous and a consistency checking approach is proposed so that all the exiting and newly defined policies are consistent with one another. The underlying formalization utilizes two formal models: 1- the user's information sharing model, and 2- the privacy-preserving model. The user's information sharing model represents all the user's information sharing activities to others. The privacy verification is performed by mapping each user's information sharing parameters (known as a state) to a state in the privacy-preserving model; a state with no mapping indicates a privacy violation. As a proof of concept, the privacy formalization and verification engine is implemented as a Java program that detects privacy violations as the user shares information in real-time. Since this framework is targeted for smart devices, which usually have low memory and low processing power, its performance was evaluated on both a PC and a Raspberry Pi model B to show the practicality of our approach.

The future work will extend the current effort to include: user privacy requirement elicitation, identification and categorization of the information shared by users, and detection of the relationship changes between a user and recipients.

The rest of the paper is organized as follows: section 2 provide the related works. Section 3 has a detailed description of our formalism and verification engine, and the implementation details of our framework are given in section 4. Moreover, the performance evaluation of the proposed framework is given in section 5. Finally, section 6 draws the conclusion of this paper and discusses the future works of our approach.

2 RELATED WORKS

For over 120 years researchers have studied privacy in different settings of technological advances [41, 45]. The first privacy theory emerged when newspapers started to publish personally intrusive articles and photographs[41]. This led to seclusion and non-intrusion theory of privacy that defined the user's privacy as "the right to be left alone" [45] or being free from intrusion [18]. As new technologies were introduced such as databases containing the personal information of the users [41] the information-related privacy concerns [38] emerged. To address these concerns researchers developed the control [46], limitation [16], and Restricted Access/Limited Control (RACL) [32] theories to enable users to control and limit their privacy while share information with others. In RACL theory, the user's privacy is implied as "a situation with regard to others [if] in that situation the individual. . . is protected from intrusion, interference, and information access by others." [42] The control, limitation and RACL theories assume a rigid definition of privacy, while in the current technological era the meaning of privacy changes based on the societal norms. To address this issue, Nissenbaum proposed the Contextual Integrity (CI) theory of privacy, [34] where privacy behaviors are affected by the context of the information sharing environment.

To implement the above theories, privacy policy languages were created based on the theories of limitation, control and RACL. The early privacy languages were either created by augmentation of access control languages or have the same structure of specifying policies as a set of access roles and information categories in a structured format like Extensible Markup Language (XML). Some well-known examples of such Languages are Platform for Privacy Preferences Project (P3P) [39], Enterprise Privacy Authorization Language (EPAL) [4], eXtensible Access Control Markup Language (XACML) [33], and Confab [19]. The early version of these languages lacked temporal modalities that were solved in the extended versions of them such as adding spatio-temporal attributes to XACML [27, 35, 43].

Another common formalism for privacy is based on transition systems where the policies are specified as action and state of information sharing. Formalizing privacy policies were based on the privacy-preserving and privacy-violating actions in the system. Also, in this formalism, the temporal characteristic of privacy was modeled using Linear Temporal Logic (LTL). Lu et al. [28] proposed a technique that translated the privacy specification of web services to LTL formulas. Then a Privacy Interface Automata (PIA) is presented to transform the messaging structure extracted from the web service business process execution language (WS-BPEL) into an automaton, creating their privacy policy model. Krishnan et al. [26] also proposed an approach to enforce privacy requirements using role-based access control and LTL. Their technique contains behavior automata that model the system behavior (gathering or using data) and an access control automata which enforce the privacy policies. Kouzapas et al. [25], combined the π-calculus and privacy calculus to verify privacy policies formally. Their framework has a type system to capture privacy related notations and a language for expressing the privacy policies. Grace et al. [17] proposed a model of user-centric privacy with a labeled transition system, which compares the cloud service privacy policies with the users' privacy preferences. However, while they provide customizable privacy preferences, they do not consider environmental variables in their model. Although this group specifies the privacy utilizing a formal semantic and considers the temporal modalities, the action based modeling of the system is not scalable [5].

The scalability issue in action based systems were addressed by Aucher et al. [5] that proposed to specify the privacy policies over the knowledge that the information sharing action exposes to the recipients of the information. In this model, privacy policy is specified as allowed and prohibited knowledge rather than actions, and different actions can result in different knowledge exchange. They used dynamic epistemic deontic logic (DEDL) as the foundation of their language. The authors define information sharing conditions as permitted or forbidden knowledge and the proposed language does not support temporal modalities. Also, Pardo et al. [36], presented a formal language for privacy policy, using epistemic logic for social network models. However, their formal privacy policy did not contain time features; later, [24, 37] extended [36] to include time characteristics to the privacy language by adding time interval and LTL which led to the creation of timed privacy framework for social media. Both frameworks used a social network model and privacy policies as properties for model checking [7] verification.

while a verity of implementation based on the theory of limitation, control and RACL continues to grow, another group of studies focused on the implementation of CI theory of privacy. Barth et al. [8] have utilized first-order logic and LTL to model the transfer of knowledge between agents during the information sharing activities that are governed by Nissenbaum's concept of *norms*. In this context, a positive norm is defined as a permission that allows information sharing activity and a negative norm prevents the information sharing activity. Further, implementation of CI was extended by DeYoung et al . [14] to include the notion of purpose and self-reference based on their Privacy Least Fixed Point (LFP) framework. The proposed framework resulted in the broader formalization of HIPAA and GLBA privacy laws.

The above approaches assume that the privacy policies will be created in a manner that are consistent with one another. However, privacy is dynamic in nature and as relationships and user's requirements changes it is required for privacy policies to change. These changes can result in privacy policy conflicts. Therefore, Breaux et al. [10] proposed Eddy that utilized CI. The goal of their research was to find privacy conflicts in multi-stakeholder privacy policies. In order to achieve that goal, natural language policies are translated to Description Logic (DL)[6] so it can be used in the formal reasoning process to investigate whether the policies are consistent. Eddy and many other frameworks that are based on CI theory are designed and develop based on the organizational privacy requirements which are not compatible with individual users privacy requirements.

For that reason, this paper defines and formalizes a user-centric privacy model utilizing CI theory. The next section describes the details on the methodology of our framework.

3 A FORMAL MODEL FOR USER-CENTRIC PRIVACY MANAGEMENT

This research extends the concept of contextual integrity [8] to provide mathematical models and algorithms that enables the creations and management of privacy norms for individual users. The extension includes the augmentation of environmental variables, i.e. time. date, etc. as part of the privacy norms, while introducing an abstraction and a partial relation over information attributes.

The proposed framework is based on two sets of formal models: 1- User's Information Sharing Model (UISM) that represents the information sharing activities in real-time, and 2- Privacy-Preserving Model (PPM) that formally specifies the user's privacy requirements. Finally, the privacy verification is performed by mapping each action in UISM to its corresponding action in the PPM. In the case of not being able to map an action a privacy violation is detected and reported to user to get confirmation. The rest of this section explains the above concepts in details.

3.1 User Information Sharing Model (UISM)

UISM is designed based on the formal definition of entities that construct *Information Communication* machanism based on agent. This is done to model user's information sharing behavior with the recipients, which are defined as agents [5, 8]. Hence, P is defined as a set of agents that are the recipient of the information sent from the user. For example, Alice and Bob are agents that the user shares information with them. In addition, T is a set of attributes that defines the information shared with $p \in P$ such as "home address" or "credit card number".

From the above definitions, a knowledge state κ is defined as a set of tuples of the form $(p, \{t_1, \dots, t_k\})$, which describes the attributes $t_i \in T$ that is shared with an agent p. For example $(Alice, \{\text{home address, credit card number}\})$ means that Alice knows about the "home address" and "credit card number". As a result, if agents have no knowledge about the user then κ can be an empty set. Therefore, the absence of tuples for p indicates that the agent p possesses no information about the user, i.e., the elements $(p, \emptyset) \notin \kappa$. Thus, κ can be defined as follows where P is a set of agents and $\mathcal{P}(T)$ is the power set of attributes,

$$\kappa \subseteq \emptyset \cup (P \times (\mathcal{P}(T) \setminus \emptyset))$$

For brevity we use $\widetilde{t}$ to represent an element of $\mathcal{P}(T)$, i.e., $\{t_1, \dots, t_k\}$.

In the proposed framework the user can perform two commands to share or stop sharing information with an agent. Each share, *sh*, or stops sharing, *st* command results in a communication action which we define as a triple $(a, p, \widetilde{t}\,)$, where $a \in \{sh, st\}$. For example, when user intend to share his/her home address with Alice, the following communication action has to be performed: $(sh, Alice, \{\text{home address}\})$. Thus, all possible communication actions can be defined as

$$Act = \{sh, st\} \times P \times (\mathcal{P}(T) \setminus \emptyset)$$

Based on the entities defined so far, the user's behavior model could be defined by a transition system where each state represents the information shared with the agents. Further, each transition is triggered by the communication action performed by the user.

Definition 1. ***(The User Information Sharing Model (UISM)*** *Let $UISMM = (K, Act, \rightarrow, \kappa_0)$ be a 4-tuple transition system where:*

- *K is a finite set of knowledge states κ.*
- *$\kappa_0 \in K$ is the initial state $\kappa_0 = \emptyset$ (no initial disclosures).*
- *Act is a set of communication actions.*
- *$\rightarrow \subseteq K \times Act \times K$ is a transition relation, transform the system state with actions $(a, p, \widetilde{t}\,)$ as follows:*
  - $\kappa \xrightarrow{(sh,p,\widetilde{t}\,)} \kappa'$, *where* $\kappa' = \kappa \cup \{(p, \widetilde{t}\,)\}$,
  - $\kappa \xrightarrow{(st,p,\widetilde{t}\,)} \kappa'$, *where* $\kappa' = \kappa \setminus \{(p, \widetilde{t}') \mid \widetilde{t} \cap \widetilde{t}' \neq \emptyset\}$.

It is important to note that the proposed model differentiates between the sequentially/simultaneously sharing of t_1 and t_2 with p. The sequential sharing results in $\kappa_1 = \{(p, \{t_1\}), (p, \{t_2\})\}$ while the simultaneous sharing results in $\kappa_2 = \{(p, \{t_1, t_2\})\}$. In κ_2 if the action $(sh, p, \{t_1, t_2\})$ occurs $(p, \{t_1, t_2\})$ is added to the new knowledge set. Thus a state contains all the three tuples $\kappa_3 = \{(p, \{t_1\}), (p, \{t_2\}), (p, \{t_1, t_2\})\}$. On the other hand, the performance of the stop command (st, p, t_2) on κ_3 will result in deletion of all the information attribute that contained t_2 from $\kappa' = \{(p, \{t_1\})\}$.
For the sequential information sharing model, we consider a scenario where user first shares his "GPS" information with Alice, second shares his "home address" with her, and third shares his billing information which is a combination of {home address, credit card number} with Alice. If the commutation action of stop sharing "home address" with Alice occurs then all the tuples that contain

"home address" like billing information will be removed from the state.

3.2 Privacy-Preserving Model (PPM)

The Privacy-Preserving Model is designed to manage and govern user's information sharing activities at run-time. Therefore, based on the proposed UISM in the previous section, PPM model is required to govern the transitions between knowledge states according to the norms that the *user* specifies.

Since in a user-centric approach is inefficient to define a separate privacy norm for each ρ (role) and τ (attribute type), the proposed model abstracts these two elements. This abstraction allows to have the same information disclosure norms with a set of agents or disclose a collection of attributes in a similar manner. For example, the user could share her current location with all transportation applications, or the user could share her credit and debit cards' numbers with her close family members. The following section describes the structure of the abstractions.

3.2.1 Abstractions and Conditions. Let $\mathcal{T}$ be a set of *attribute types* and let AT be a partial map $AT : \mathcal{P}(T) \mapsto \mathcal{T}$. That is, AT maps $\widetilde{t}$ to an attribute type $\tau \in \mathcal{T}$. We can impose a partial order $\preceq$ on τ based on the subset relation between AT's domain elements $\widetilde{t}$. We say that $\tau_1 \preceq \tau_2$ if there are exist $\widetilde{t_1}$ and $\widetilde{t_2}$ such that $AT(\widetilde{t_1}) = \tau_1$, $AT(\widetilde{t_2}) = \tau_2$ and $\widetilde{t_1} \subseteq \widetilde{t_2}$.

Figure 1a, and 1b demonstrate an example of hierarchy structure and some attributes and attribute types in that structure. The dashed line represents the mapping between an attribute and its type and the solid lines depict the order relation between the attribute and types.

Similar to [8] that defines the concept of role abstraction, we define a set of *agent roles* $\mathcal{R}$ that can be assigned to an agent p. An agent can be assigned to multiple roles and roles are partially ordered based on their implication relation of their semantics.

In this paper, the partial order $\leq$ on $\mathcal{R}$ is predefined as an input to the model, such that the role, ρ_1, "close friend" implies the role, ρ_2, "friend", i.e., $\rho_2 \leq \rho_1$. The order between roles implies the amount of relative privacy restriction of them where $\rho_2 \leq \rho_1$ means that ρ_2 is more restrictive compared to ρ_1.

In this approach each agent must be associated with at least one role. Thus, we define the agent role as a function AR that maps an agent to a nonempty set of roles: $AR : P \mapsto \mathcal{P}(\mathcal{R}) \setminus \emptyset$. When role ρ is assigned to an agent p, then the systems adds additional roles that related to ρ through $\leq$. In other words, the set of roles for p should be closed under $\leq$. For example, if the agent p is assigned the role "close friend" ρ_1, then the system adds "friend" role ρ_2 to p as well, resulting in $AR(p) = \{\rho_1, \rho_2\}$.

For brevity to show the roles and information attributes that have a common child but are not in a partial relation with each other we use the $< child >$ notation as follow:

(1) $\rho_1 < p > \rho_2 = \exists p \in \mathcal{P} : \rho_1 \in AR(p) \wedge \rho_2 \in AR(p) \wedge \rho_1 \npreceq \rho_2 \wedge \rho_2 \npreceq \rho_1$
(2) $\tau_1 < t > \tau_2 = \exists \widetilde{t} \in \mathcal{P}(T) : AT(\widetilde{t}) \preceq \tau_1 \wedge AT(\widetilde{t}) \preceq \tau_2 \wedge \tau_1 \npreceq \tau_2 \wedge \tau_2 \npreceq \tau_1$

Using these abstractions the user can define ***access permissions*** $\mathcal{A}$ as a subset of $\mathcal{R} \times \mathcal{T}$ such that if an element $(\rho, \tau) \in \mathcal{A}$ then all agents with role ρ are allowed to access attributes with type τ.

The above abstractions of roles and information attributes provide a better flexibility in defining privacy norms. However, this definition is not complete yet, as it does not take into the consideration the environmental conditions where the information is disclosed to the recipients and has no sensitivity over the patterns and sequence of the information disclosure. Imagine, user is interested in restricting access of agents in ρ role to its attribute type τ to a particular time interval during a work day. Moreover, the user might allow only up to two (ρ, τ) accesses per such interval.

In order to overcome this limitation, our formalism introduces the logic for environmental conditions ψ and temporal conditions φ to the definition of the privacy norm. In this model, environmental conditions are represented a set of variables V, where each $v \in V$ describes the state of an environment such as system's time, day and other attributes. Then, V is partitioned into subsets V_i by variables' type like integers, boolean, reals and so on. It is assumed that each type has a set of predicates $Pred_i$ and set of syntax rules to construct such predicates from the variables and non-logical symbols, e.g., constants. Then an environmental condition (Ψ) is expressed as a propositional logic over those predicates and variables, i.e., $v \in V_i$, $pred_i \in Pred_i$ as follows:

$$\psi ::= \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid pred_i, \forall V_i \in V$$

While $Pred_i$ could be produced by an arbitrary complex yet decidable theory for the data type such as Presburger arithmetic for integers, we argue that less complex theories could be adequate[3]. For example, for integer environmental variables V_I and boolean V_B environmental variables the following grammar could be sufficient to express basic and easily comprehensible predicates $pred_i$:

$$pred_I ::= v \leq n \mid v < n \mid v == n, v \in V_I, n \in \mathbb{Z}$$
$$pred_B ::= v \mid true \mid false, \ v \in V_B$$

The next entity that is defined as part of the privacy norm is the temporal condition φ. In order to keep the conditions flexible and generic, we utilize temporal logic expressions to describe temporal features of the privacy requirements. While Linear Temporal Logic (LTL) is very popular in expressing broad range of liveness conditions, they are difficult to read and understand. Utilizing LTL requires a strong mathematical background, and is cumbersome for an average system modeler to implement. Further, for the purpose of defining temporal conditions in privacy norm a simplified grammar will suffice, i.e define the precedence of two communication actions or a constant occurrence a communication actions can be sufficiently defined by the concatenation and Kleen star operations over $\mathcal{A}$ (the alphabet):

$$\varphi, \phi ::= (\rho, \tau) \mid \varphi \cdot \phi \mid \varphi^*, (\rho, \tau) \in \mathcal{A}$$

The Φ notation is used to represent a set of φ, in which each φ for a given role ρ, can be expressed as a regular expression that allows sharing attributes of type τ_2 after the sharing of attributes of type τ_2 as follows:

$$\varphi = \mathcal{A}_1^* \cdot ((\rho, \tau_1) \cdot \mathcal{A}_1^* \cdot (\rho, \tau_2))^* \cdot \mathcal{A}_1^*$$

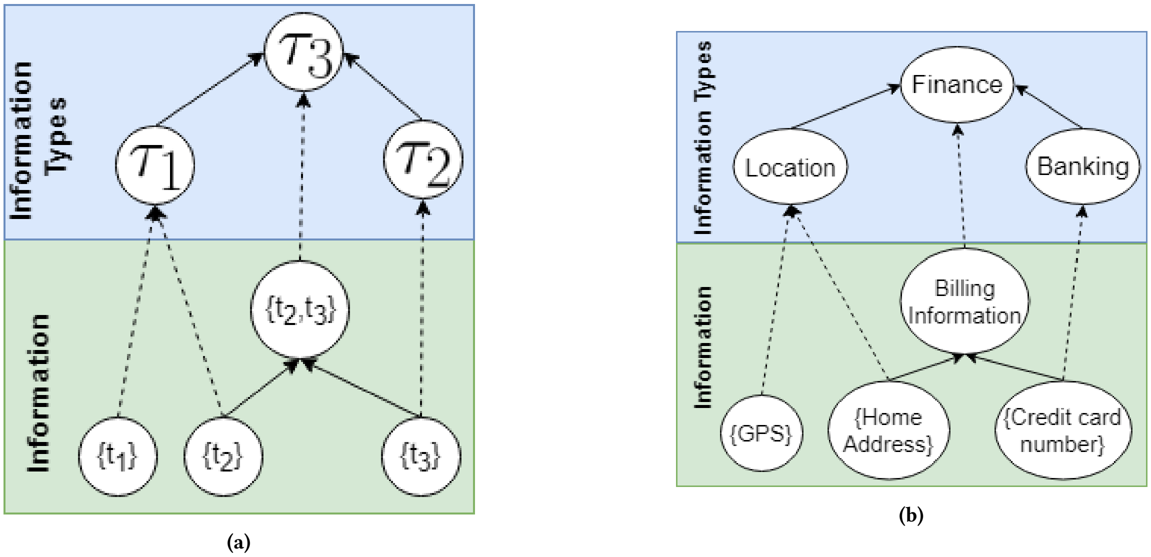


Figure 1: (a) An example of the partial order of the attributes and attribute types where the top layer show the attribute types and the bottom layer show the information themselves. (b) t_1 =GPS information, t_2 = home address, and t_3 = credit card number. The middle layer represents the information that are used together for example the credit card number and the home address go together for billing information that is a considered as financial type.

Here $\mathcal{A}_1 = \mathcal{A} \setminus \{(\rho, \tau_1), (\rho, \tau_2)\}$ In addition, the repetition of an event up to a constant k times could be expressed with the following formula, where the power operator describes the number of times a regular expression should be repeated.

$$\varphi = \mathcal{A}_2^*((\rho, \tau) \cdot \mathcal{A}_2^*)^k$$

where $\mathcal{A}_2 = \mathcal{A} \setminus \{(\rho, \tau)\}$.

Now that we have defined each elements in the privacy norm, the next section describes the formal specification of the privacy norm and techniques to ensure the consistency of the privacy requirements.

3.2.2 *Norms and their Consistency.* In this research, norms are the formal definition of user's privacy requirements that are used to govern user's information sharing behavior. In order to minimize the risk of unwanted information sharing, we assume that if an action is not explicitly defined as part of the user's privacy policies then it is forbidden. Therefore, the only type of norms that the user defines are positive norms, i.e., *allowed* norms.

In this context norm is formulated as a relation between access permission, environmental, and temporal conditions. Hence, norm is represented as a tuple $((\rho, \tau), \psi, \varphi,)$, where $(\rho, \tau) \in \mathcal{A}$ and $\psi \in \Psi$, $\varphi \in \Phi$. The first element of the tuple represents the privacy policy, while the second and the third elements of the tuple describe the conditions under which the transfer of information should occur. The set of such is referred to as a set of norms $\mathcal{N}$.

The set $\mathcal{N}$ has the uniqueness property, that is, only one tuple with the given (ρ, τ) values is allowed in the set. However, the uniqueness property is not sufficient to ensure the consistency of the privacy norms due to the partial relations that exist among the roles and attribute types. Thus, in order to utilize $\mathcal{N}$ for privacy management and detection of information disclosure, a consistency check is required. The Table 1 demonstrates a detailed explanation with examples of the different possible cases of role and attributes types that two norms can have during consistency checking. The row headers show the roles and the column headers show the attribute types. The cells in gray are the example of their above conditions.

Definition 2. ***(Consistent Norms)*** *Two norms $n_1 = ((\rho_1, \tau_1), \psi_1, \varphi_1)$ and $n_2 = ((\rho_2, \tau_2), \psi_2, \varphi_2)$ are consistent when one of the four consistency conditions holds:*

C1. $\nexists p \in \mathcal{P} : \rho_1 \in AR(p) \wedge \rho_2 \in AR(p)$, that is, the norms defined for the roles with no common agents. (Table 1 row G)

C2. $\nexists \widetilde{t} \in \mathcal{P}(T) : AT(\widetilde{t}) \leq \tau_1 \wedge AT(\widetilde{t}) \leq \tau_2$, that is, norms are defined for attribute types with no common information attribute.(Table 1 column 5)

Before defining the last two conditions of consistency, we propose some limitations over the access permission and sequencing conditions of the privacy norms. Since both of these elements are defined for a specific roles and attribute type parameters, the first restriction is defined over the roles so that the same role should be used in the access permission and the sequencing condition of a norm. In the absence of this restriction, it is possible to create two norms that have a consistent sequencing condition but inconsistent access permission or vice versa. In addition, this restriction enforces a constant role across the regular expression of the sequencing condition that reduces the regular expression's complexity by eliminating the need for a homomorphic function over the roles. The second restriction is defined over the attribute types, $\forall \tau \in \varphi \quad \tau_i \nleq \tau_j \quad 0 \leq i, j \leq n$ (An attribute type and its children

Table 1: The possible consistency cases based on the roles and information attribute types relations and the constrains over the conditions that result in consistency. The notations *Fr*=Friends, *BFr*=Best Friends, *CoWr*=Co-Workers, *Fml*=Family, *Loc*=Location, *Fin*=Finance, *Hlth*=Health, and *Bank*=Banking information

		1	2	3	4	5
		$\tau_1 < \tau_2$	$\tau_2 < \tau_1$	$\tau_1 = \tau_2$	$\tau_1 < e > \tau_2$	$\tau_1 < none > \tau_2$
		Loc < Fin	*Loc < Fin*	*Loc = Loc*	*Fin < Loc > HLth*	*Loc < none > Bank*
A	$\rho_1 < \rho_2$	$c_2 \Leftrightarrow c_1$ $\mathcal{L}(s_1) = \mathcal{L}(s_2)$	$c_2 \implies c_1$ $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$	$c_2 \implies c_1$ $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$	$c_2 \implies c_1$ $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$	True
B	*Fr < BFr*	Share Loc with Fr when c1 an s1, share Fin with BFr when c2 and s2. Fin should be guarded the same or better, $c_1 \implies c_2$, $\mathcal{L}(s_2) \subseteq \mathcal{L}(s_1)$. BFr can have less restrictive access, $c_2 \implies c_1$, $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$	Share Fin with Fr when c1 and s1, share Loc with BFr when c2 and s2. Fin should be guarded the same or better, $c_2 \implies c_1$, $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$. BFr can have less restrictive access $c_2 \implies c_1$, $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$	Share Loc with Fr when c1 and s1, sare Loc with Bfr when c2 and s2. Loc should be guarded at least the same way, $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$. BFr can have less restrictive conditions, $c_2 \implies c_1$, $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$	Share Fin with Fr and Health with BFr (or vice versa) which can share Loc. Loc should be guarded at least the same way $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$. BFr can have less restrictive condition, $c_2 \implies c_1$, $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$	Since Loc and Bank are incomparable then those norms should always be consistent.
C	$\rho_1 = \rho_2$	$c_1 \implies c_2$ $\mathcal{L}(s_2) \subseteq \mathcal{L}(s_1)$	$c_2 \implies c_1$ $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$	False	$c_2 \Leftrightarrow c_1$ $\mathcal{L}(s_1) = \mathcal{L}(s_2)$	True
D	*Fr = Fr*	Share Loc with Fr when c1 and s1, share Fin with Fr when c1 and s2. Fin should be guarded the same or better way $c_1 \implies c_2$, $\mathcal{L}(s_2) \subseteq \mathcal{L}(s_1)$. Fr should have at least the same access, $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$.	Share Fin with Fr when c1 and s1, share Loc with Frien when c2 and s1. Fin should be guarded the same or better way, $c_2 \implies c_1$, $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$. Fr should have at least the same access $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$	There should be only one rule for the same role and attribute type - the uniqueness property	Share Fin with Fr when c1 and s1, share Health with Fr when c2 and s2, which can share the same attribute Loc. Loc should be guarded at least the same way $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$. Fr should have the same access $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$	Since Loc and Bank are incomparable then those norms should always be consistent.
E	$\rho_1 < p > \rho_2$	$c_1 \implies c_2$ $\mathcal{L}(s_2) \subseteq \mathcal{L}(s_1)$	$c_2 \implies c_1$ $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$	$c_2 \Leftrightarrow c_1$ $\mathcal{L}(s_1) = \mathcal{L}(s_2)$	$c_2 \Leftrightarrow c_1$ $\mathcal{L}(s_1) = \mathcal{L}(s_2)$	True
F	Fr Anna CoWr	Share Loc with Fr when c1 and s1, share Fin with CoWr when c2 and s2, which have Anna as a common agent. Fin should be guarded the same or better way $c_1 \implies c_2$, $\mathcal{L}(s_2) \subseteq \mathcal{L}(s_1)$. Fr and CoWrk should have at least the same access to Loc $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_2) = \mathcal{L}(s_1)$, since they share an agent.	Share Fin with Fr when c1 and s1, share Loc with CoWrk when c2 and s2, which have Anna a common agent. Fin should be guarded better than Loc $c_2 \implies c_1$, $\mathcal{L}(s_1) \subseteq \mathcal{L}(s_2)$. Fr and CoWrk should have at least the same access to Loc $c_2 \Leftrightarrow c_1$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$, since they share an agent.	Share Loc with Fr when c1 and s1, share Loc with CoWrk, when c1 and s2, which have Anna as a common agent. Loc should be guarded the same way $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$. Fr and Cowrk should have the least the same access to Loc, $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$, since they share an agent.	Share Fin with Fr when c1 and s1, share Health with CoWrk when c2 and s2, which have Anna as a common agent. Loc should be guarded at least the same way $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$. Fr and CoWrk should have the same access to Loc $c_1 \Leftrightarrow c_2$, $\mathcal{L}(s_1) = \mathcal{L}(s_2)$, since they share an agent.	Since Loc and Bank are incomparable then those norms should always be consistent.
G	$\rho_1 < none > \rho_2$	True	True	True	True	True
H	Fr, none, Fml	Since Fr and Fml are incomparable then those norms should always be consistent.	Since Fr and Fml are incomparable then those norms should always be consistent.	Since Fr and Fml are incomparable then those norms should always be consistent.	Since Fr and Fml are incomparable then those norms should always be consistent.	Since Fr and Fml are incomparable then those norms should always be consistent.

are not allowed to exist in the same regular expression). This restriction ensures that all the communication actions are inspected not only for the super-type τ, that is explicitly inferred from the communication action, but also for all the children of τ that will be implicitly revealed by that communication action. Without this restriction, it is possible to create a regular expression that allows for sharing an attribute type and its children consecutively while it is not taking into the account that the children are shared more than once.

Further, the comparisons of the access permission component of the norms are conducted based on the partial relations that exists over the roles and attribute types. In addition, the comparison between the environmental conditions is implemented based on the Boolean algebra. To examine the sequencing conditions for consistency, we need to compare the regular expressions. the comparison of two regular expressions is not possible if they do not share the same alphabet. Therefore, we need to introduce a mechanism that projects the language of one regular expression to the other one and brings the regular expressions to a common alphabet.

Definition 3. ***(Projection of the Language)*** *Let φ_1 and φ_2 have the following symbols to be tracked:*

$$\varphi_1 = \{(\rho, \tau_1), (\rho, \tau_2), \ldots, (\rho, \tau_k)\}$$

$$\varphi_2 = \{(\rho', \tau'_1), (\rho', \tau'_2), \ldots, (\rho', \tau'_n)\}$$

We define $\widetilde{\varphi_1} = \mathcal{L}_{\downarrow}(\varphi_1)_{\varphi_2}$ as the projection of φ_1 on φ_2 where $\mathcal{L}_{\downarrow}$ receives a regular expression and maps it to another one. To achieve a similar language to compare φ_1, φ_2 we traverse over the attribute types. For each attribute type, we check for its children or another attribute type that has a common child in the other regular expression and add the children or the common child to a set in a map. After traversing over all the attribute types in both φ_1, φ_2 to substitute the uncommon parts, we generate all the possible substitution for attribute type τ_i exist in the map. The substitution for τ_i for reaching a common language is a disjunctive regular expression. The disjunctive regular expression is generated as follows. Let sub be a set of all τ_i children and common children that have been found in the other regular expression. We define $\widetilde{sub} = \mathcal{P}(sub) \setminus \emptyset$. For each $s \in \widetilde{sub}$ we generate all the permutations of elements of s and add them to the regular expression with disjunction operator. For example, $sub = \{\tau_a, \tau_b\}$ then $\widetilde{sub} = \{\{\tau_a\}, \{\tau_a\}, \{\tau_a, \tau_b\}\}$ and the result of the regular expression that is used for substitution is $\tau_a|\tau_b|\tau_a\tau_b|\tau_b\tau_a$. After reaching the same alphabet, the consistency of the regular expressions can be decides based on the norms' access permission.

C3 . $\rho_1 < \rho_2$ and either $\tau_1 \leq \tau_2$ or $\tau_2 \leq \tau_1$ then $\psi_1 \implies \psi_2 \wedge \mathcal{L}_{\downarrow}(\varphi_1)_{\varphi_2} \subseteq \mathcal{L}_{\downarrow}(\varphi_2)_{\varphi_1}$, that is, n_2 is for a specialized role ρ_2 of ρ_1 and its attribute type τ_2 encompasses τ_1 or vise verse then environmental condition of ψ_2 should be the same or less restrictive than of ψ_1 and its regular expression φ_2 should describe the same or less restricted projected language than of φ_1.(Table 1 row A,C and columns 1,2,3)

C4 . $\rho_1 < p > \rho_2$ *or* $\tau_1 < t > \tau_2$ *then* $\psi_1 \Leftrightarrow \psi_2 \wedge \mathcal{L}_{\downarrow}(\varphi_1)_{\varphi_2} = \mathcal{L}_{\downarrow}(\varphi_2)_{\varphi_1}$. *If there is at least one agent that can be assigned to both unrelated roles or an information attribute that share a common child then the environmental conditions and the projected language of the regular expressions must be equivalent.(Table 1 row E and columns 4)*

3.2.3 Policy Compliance Verification. The set of norm $\mathcal{N}$ defines a Privacy-Preserving Model, (PBM) which describes compliant information communication actions at the level of attribute type and agent role abstraction levels. The knowledge states of PBM are consists of tuples (ρ, τ), which indicate that at least one agent with ρ role know about attribute represented by τ. The transitions represent the abstracted communication actions $\widehat{Act}$ from $\{sh, st\} \times \mathcal{R} \times \mathcal{T}$ guarded by conditions Φ and Ψ defined in $\mathcal{N}$.

Definition 4. ***(Privacy-Preserving Model)*** *is a set of observers over norms* $\mathcal{N}$ *where each observer is a tuple of* $(\widehat{K}, \widehat{Act}, c, m)$ *representing* $n_i = ((\rho, \tau), \psi, \varphi) \in \mathcal{N}$ *where* $\widehat{K} = (\rho, \tau)$, $c = \psi$ *is the pre-condition and m is a monitor representing* φ *regular expression. The transition* $\widehat{Act}$ *is given to Monitor m to update the state of the monitor.*

3.2.4 Verification. To ensure that the user's behavior is compliant with the privacy policy, we need to map the current state and the next state of user's behavior model to the privacy preserving behavior model.

Definition 5. ***(Mapping from user behavior to privacy preserving domain)*** *Let* $MS : K \longrightarrow \widehat{K}$ *be a surjective function, where* $MS(p, \widehat{t}) = \{(\rho, \tau) | \rho = AR(p), \tau = AT(\widetilde{t})\}$ *and* $MT : Act \longrightarrow \widehat{Act}$ *where:*

$$MT(a, p, t) = \{(\rho, \tau) | \rho \in AR(p) \wedge \tau \in AT(t)\}\ if\ a = sh$$

In the case that there is no mapping for the next state in the PPM, the communication action that triggered that transition will be reported to the user as disclosing.

Definition 6. ***(Valid user behavior)*** *Let user behavior system be at state k that maps to* $\widehat{k}$ *in the privacy preserving behavior model and the action* (sh, p, t) *happens. If* $MP(p, t)$ *exists, and the environmental variables satisfy* ψ *and* $m(MS(a, p, t))$ *is in the final state then the communication action Act is valid.*

The goal of privacy rules is to prevent the user from entering into a privacy violating states.

After reporting a privacy-violating action the user can ignore it and the framework allow the information sharing to happen. All this communication happens through the user interface of the framework. The next section provides implementation details of the framework's components.

4 IMPLEMENTATION

As a proof of the concept, we prototyped the proposed framework in the Java programming language [1]. Figure 2 depicts a diagram of the implementation's architecture. The blue components show the libraries and technologies used in the proposed framework. The proposed framework is modularized into three layers:(1) User interface layer which takes the user's intentions in a structured format,(2) Translation layer which translates the frameworks from UI to privacy norms and formal notation, (3) Verification layer that evaluates norms consistency and compliance of the information sharing action with privacy norms.The following sections describe the implementation details of each of the components in each layer.

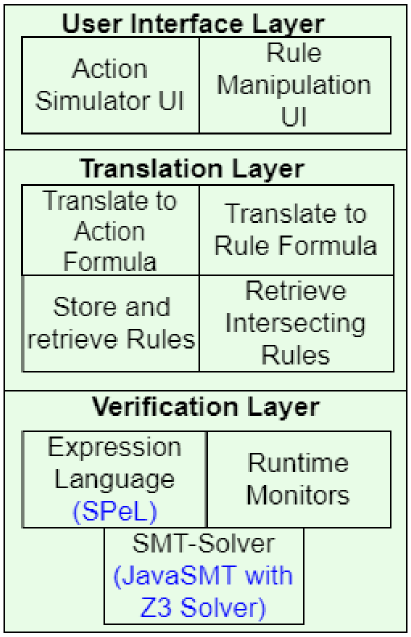

User Interface Layer
Action Simulator UI
Rule Manipulation UI
Translation Layer
Translate to Action Formula
Translate to Rule Formula
Store and retrieve Rules
Retrieve Intersecting Rules
Verification Layer
Expression Language (SPeL)
Runtime Monitors
SMT-Solver (JavaSMT with Z3 Solver)

Figure 2: The architecture of user-centric privacy framework.

4.1 User Interface Layer

The user interface (UI) layer facilitates interactions between the user and the proposed framework. Through the UI the user can add and view the existing privacy norms and get privacy violation reports. The UI is designed to conceal the complexity of the underlying formalism and verification from the user. The UI hides the complexities by allowing the users to express their privacy intentions as a structured input. Using the UI the user can select the role and attribute type from a drop-down list. To create the environmental conditions, the user can provide arbitrary inputs for environmental variables or choose between predefined conditions e.g., daytime, nighttime, weekends. Also, the user can specify the desired information sequence in the form of precedence or repetition templates like “X happens after Y" or “X happens k times”. These templates will be translated to sequencing conditions.

4.2 Translation Layer

The translation layer receives the structured input from the UI and translates it into formal notation. The formal notations and maps described in the methodology section can be implemented as tables in a database. The norm are stored in the norms table where the table attributes are the role, attribute type, the environmental conditions, and the DFA state of the sequencing conditions. The primary key of the norms table is the pair of (ρ, τ). The system

[1] https://github.com/wxyzabc/UserCentricPrivacy

queries the database to retrieve the norms in order to either verify an action or check the consistency of a new norm. To evaluate each action with the attribute t and the agent p, norms that have roles where $\rho = AR(p)$ and attribute type $\tau = AT(t)$ will be retrieved from the norm table and sent to the verification layer.

4.3 Verification Layer

This layer verifies the information sharing actions compliance with the privacy norms and the consistency of a new norm with existing norms. If an information sharing action violates the privacy norms or a new norm causes inconsistency, then this layer sends a violation report to the UI to inform the user. The user can ignore the violation caused by the information sharing action and allow the information to be shared. With an inconsistent norm, the user has to change the new norm so that it will be consistent with other norms. The rest of this section describes the verification method of information sharing actions and privacy norms in more detail.

4.3.1 Verification of norms for Inconsistency. When a new norm is created, the framework checks the consistency of the new norm with the existing norms. Based on the consistency constraints in section 3.2.2 the framework first ensures that the new norm access permission does not exist in the database. Then the new norm's environmental conditions are checked for consistency. The framework parses the string of the environmental conditions and changes them to SMT solver formulas. Then the SMT solver needs to prove that the implication or equivalency relation holds and it is always valid. Validation assessment of formula f by SMT solvers is done by proving that $\neg f$ is unsatisfiable, hence f always evaluates to true. By proving that there is no combination of variables that satisfy $\neg f$ it can be concluded that f is a tautology. In a case that the solver finds a solution to $\neg f$, the user is asked to change the inconsistent new norms. Further, since efficiency is important in real-time systems, we need to assign a time limit for the solver. If the solver times out or returns UNKNOWN the user will be notified. Finally, if the norm was consistent it will be added to the database. The implementation of the proposed framework utilizes JavaSMT [23] with the Z3 solver version 4.3.2 [13] for consistency checking over the environmental variables and "brics" library version 1.12-1 [31] for sequencing conditions.

4.3.2 Verification of Actions for violation. For each action (sh, p, t) , the framework finds the attribute type of t and the role of p. Then the privacy norms tables are queried to find the norms with the access permission $(AR(p), AT(t))$ as their primary key. If the query returns no results, it means that no norm allows sharing information t with agent p. However, If the query returns results, it indicates that there exists a mapping from a state in UBM to a state in the PPM. Then the framework checks for the satisfaction of the environmental conditions and sequencing conditions before taking the transition to the mapped state.

Since the norm conditions are dynamic, they cannot be hard-coded in the verification engine. Therefore to check the environmental variables a mechanism is needed to enable the verification engine to handle change in the conditions. Therefore, the conditions are formed and evaluated at run-time based on the stored environmental constraints in the database. For the implementation of such a mechanism that allows for dynamic manipulation and evaluation of conditions, the Expression Languages (EL) can be used. EL receives an object and a logical expression as a string and evaluates whether the object properties satisfy the expression or not. In our implementation,the current snapshot of the environment is given to the EL as the input object that has the environmental values and the EL expression string is the environmental constraints of the retrieved privacy norms. This framework employs Spring Expression Language (SpEL) [1] as the EL library. EL only checks for the satisfaction of the environmental conditions and if they are not satisfied then the transition guard is not satisfied. Therefore, the action violates the privacy model. However, if the environmental conditions are satisfied then we check for the satisfaction of sequencing conditions.

Sequencing conditions implemented as run-time monitors from the regular expressions stored in the database. A run-time monitor is a deterministic finite automaton (DFA) that is created based on a regular expression. The DFA representing the sequencing condition has a pointer to its current state and changes its state with the occurrence of information sharing actions. If the new state in the DFA monitor is not a final state, then the action is not valid, and the system reports the violation to the user. Different libraries exist for creating run-time monitors such as AspectJ, but the monitors created by them are static. Therefore, a change in one of the regular expressions demands a reset in all the monitors. In the proposed framework the regular expressions are dynamic, and changing a regular expression only causes a reset in the corresponding DFA. Another method for implementing the sequencing conditions is to store a history of information sharing actions; however, with each information sharing action, the history will grow, and to potentially infinite size. With the run-time monitors, the number of the DFAs are constant and equal to the number of the norms with sequencing conditions. Algorithm 1 shows the general steps taken to implement the information sharing action verification process.

Algorithm 1: Action verification algorithm.

```
Input: CA (Communication action)
Output:Boolean value indicating the verification result.
norms=[]
roles=CA.recipient.getRoles()
types=CA.informationAttribute.getType()
for r in roles and t in types do
    norms.append(getnorms(r,t))
if (norms.size> 0) then
    for j in norms do
        if !(j.evalEnvironmentalCondition (CA.environment)) then
            return false
        else
            if !(j.evalSequencingCondition(CA)) then
                return false
    return true
return false
```

Considering the above implementation, in the next section we discuss the performance evaluation of the proposed framework.

5 PERFORMANCE EVALUATION

The proposed framework is designed for user-centric applications; therefore, it should have acceptable performance on smart devices such as smart-phones, internet of things devices and etc. The main challenge in this area is that usually, these devices have low memory and computational power. Since detection of privacy violations in such applications supposed to be real-time, a framework with a substantial performance overhead cannot deliver the desired results. Therefore, our implementation was tested for performance evaluation on a Raspberry Pi model B with 700 MHz CPU, 512 MB RAM and running Raspbian 4.9 operating system. As well as a PC with 3.0 GHz AMD Phenom II X4 945 processor, with 8 GB of memory and Windows 7 operating system. The privacy policy created for this test contained 81 privacy norms over 12 attribute types and 16 roles which 8 of them have nonempty intersections with another groups.

Table 2 shows the results of the information sharing action verification performance evaluation. The number in each column indicates the average verification response time for each part of the a privacy norm. The average was computed for 20 information sharing actions which half were privacy violating actions and the other half were non-violating actions. Also, notice that the performance of the action verification depends on the performance of underlying database software and expression language library. In the implementation of our framework, we used MariaDB version 10.2 database and SpEL 3.1.0 as the EL library.

Table 2: Action verification performance evaluation results. The columns show the response time for Access Permission (AP), Environmental Conditions (EC), Sequencing Conditions (SC).

Machine	Action Verification		
	AP	EC	SC
PC	1.5 ms	0.5 ms	3.5 ms
Pi B	39 ms	6 ms	540 ms

The average time for the consistency check performance evaluation on the PC was 39 ms and for Raspberry Pi model B was 849 ms. Also, notice that the performance of this consistency checking depends on the performance of the underlying solver and the domain size of the environmental variables (since the solvers are faster when the search domain is smaller). For example, in our implementation, the norms time conditions were specified as (hours×100+minutes) and time intervals could be subsets of each other. Table 3 shows the SMT-solver performance for constraints with 5,10,20,50,100, and 500 environmental variables. The over-head of bric library for language sunset and equivalency is around 7ms on average. However, the projection algorithm is the bottleneck since it computes the permutation of the information types that are needed for substitution in the regular expression. Due to this drawback the framework limits the number of children for each attribute to 5 children.

Table 3: Performance of consistency checking for Environmental Variables

Number of Variables	5	10	20	50	100	500
Implication time (ms)	26	28	30	40	35	66
Equivalency time (ms)	32	34	35	46	41	67

6 CONCLUSION AND FUTURE WORKS

Administrating and managing users' privacy is a major challenge in the digital age. Privacy has a different meaning to different users depending on their personality, age, social status, cultural background, and many other factors. However, current privacy management systems cannot address these privacy needs adequately since they are not designed based on the users' privacy perspectives. Therefore, the lack of user-centric privacy management tools and algorithms limits users' ability to have control over their data sharing activities and puts unaware users at risk of information disclosure. In order to overcome these limitations, the proposed framework provides a privacy formalism and verification engine to specify and model privacy from the user's perspective. Moreover, as a proof of concept, a framework was implemented and tested based on the described formalism. In the proposed model, the contextual integrity theory has been customized to address the privacy needs of individual users. Further, the user-centric privacy framework is meant to be utilized in the new generation of smart devices and IoT, which compared to servers and general purpose computers, have lower memory and computational power. These limitations justify the use of regular expressions instead of Linear Temporal Logic (LTL) in our paper since empirical evidence [9] shows that the evaluation of the regular expressions has significantly less overhead compared to LTL.

The future work will eliminate the current user interface and user's privacy norms will be generated automatically utilizing text analysis, speech recognition, and AI algorithms that can infer user's privacy policies based on the user's relationships and information sharing behaviors.

ACKNOWLEDGMENTS

The authors would like to thank National Science Foundation for its support through the Computer and Information Science and Engineering (CISE) program and Research Initiation Initiative(CRII) grant number 1657774 of the Secure and Trustworthy Cyberspace (SaTC) program: A System for Privacy Management in Ubiquitous Environments.

REFERENCES

[1] [n. d.]. Spring Expression Language. https://docs.spring.io/spring/docs/3.0.x/reference/expressions.html. Accessed: August 31, 2018.

[2] Alessandro Acquisti, Laura Brandimarte, and George Loewenstein. 2015. Privacy and human behavior in the age of information. *Science* 347, 6221 (2015), 509–514.

[3] Alessandro Acquisti and Jens Grossklags. 2005. Privacy and rationality in individual decision making. *IEEE security & privacy* 3, 1 (2005), 26–33.

[4] Paul Ashley, Satoshi Hada, Günter Karjoth, Calvin Powers, and Matthias Schunter. 2003. Enterprise privacy authorization language (EPAL). *IBM Research* (2003).

[5] Guillaume Aucher, Guido Boella, and Leendert Van Der Torre. 2011. A dynamic logic for privacy compliance. *Artificial Intelligence and Law* 19, 2-3 (2011), 187.

[6] Franz Baader, Ian Horrocks, and Ulrike Sattler. 2008. Description logics. *Foundations of Artificial Intelligence* 3 (2008), 135–179.

[7] Christel Baier, Joost-Pieter Katoen, and Kim Guldstrand Larsen. 2008. *Principles of model checking.* MIT press.

[8] Adam Barth, Anupam Datta, John C Mitchell, and Helen Nissenbaum. 2006. Privacy and contextual integrity: Framework and applications. In *Security and Privacy, 2006 IEEE Symposium on*. IEEE, 15–pp.

[9] Ilan Beer, Shoham Ben-David, and Avner Landver. 1998. On-the-fly model checking of RCTL formulas. In *International Conference on Computer Aided Verification*. Springer, 184–194.

[10] Travis D Breaux, Hanan Hibshi, and Ashwini Rao. 2014. Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. *Requirements Engineering* 19, 3 (2014), 281–307.

[11] Carole Cadwalladr and Emma Graham-Harrison. 2018. Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. *The Guardian* 17 (2018).

[12] Federal Trade Commission et al. 2012. Recommendations for Businesses and Policymakers. *Washington, DC (http://www. ftc. gov/sites/default/files/documents/reports/federal-trade-commission-report-protecting-consumer-privacy-era-rapid-change-recommendations/120326privacyreport. pdf)* (2012).

[13] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient SMT solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 337–340.

[14] Henry DeYoung, Deepak Garg, Limin Jia, Dilsun Kaynar, and Anupam Datta. 2010. Experiences in the logical specification of the HIPAA and GLBA privacy laws. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*. ACM, 73–82.

[15] Michael D Ekstrand, Rezvan Joshaghani, and Hoda Mehrpouyan. 2018. Privacy for All: Ensuring Fair and Equitable Privacy Protections. In *Conference on Fairness, Accountability and Transparency*. 35–47.

[16] Ruth Gavison. 1980. Privacy and the Limits of Law. *The Yale Law Journal* 89, 3 (1980), 421–471.

[17] Paul Grace and Michael Surridge. 2017. Towards a model of user-centered privacy preservation. (2017).

[18] Jamal Greene. 2009. The So-Called Right to Privacy. *UC Davis L. Rev.* 43 (2009), 715.

[19] Jason I Hong and James A Landay. 2004. An architecture for privacy-sensitive ubiquitous computing. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*. ACM, 177–189.

[20] White House. 2015. Administration discussion draft: Consumer Privacy Bill of Rights Act of 2015. *Retrieved on November* 15 (2015), 2015.

[21] Rezvan Joshaghani, Michael D. Ekstrand, Bart Knijnenburg, and Hoda Mehrpouyan. 2018. Do Different Groups Have Comparable Privacy Tradeoffs? *At Moving Beyond a One-Size Fits All Approach: Exploring Individual Differences in Privacy, a workshop at the ACM Conference on Human Factors in Computing Systems (CHI)* (2018).

[22] Rezvan Joshaghani and Hoda Mehrpouyan. 2017. A Model-Checking Approach for Enforcing Purpose-Based Privacy Policies. In *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, 178–179.

[23] Egor George Karpenkov, Karlheinz Friedberger, and Dirk Beyer. 2016. JavaSMT: A unified interface for SMT solvers in Java. In *Working Conference on Verified Software: Theories, Tools, and Experiments*. Springer, 139–148.

[24] Ivana Kellyérová. 2017. A Real-Time Extension of the Formal Privacy Policy Framework. (2017).

[25] Dimitrios Kouzapas and Anna Philippou. 2017. Privacy by typing in the pi-calculus. *arXiv preprint arXiv:1710.06494* (2017).

[26] Padmanabhan Krishnan and Kostyantyn Vorobyov. 2013. Enforcement of privacy requirements. In *IFIP International Information Security Conference*. Springer, 272–285.

[27] Ki Young Lee, Aleum Kim, Ye Eun Jeon, Jeong Joon Kim, Yong Soon Im, Gyoo Seok Choi, Sang Bong Park, Yun Sik Lim, and Jeong Jin Kang. 2015. Spatio–temporal XACML: the expansion of XACML for access control. *International Journal of Security and Networks* 10, 1 (2015), 56–63.

[28] Jiajun Lu, Zhiqiu Huang, and Changbo Ke. 2014. Verification of Behavior-aware Privacy Requirements in Web Services Composition. *JSW* 9, 4 (2014), 944–951.

[29] Mary Madden, Aaron Smith, and Jessica Vitak. 2007. Digital Footprints: Online identity management and search in the age of transparency. (2007).

[30] Hoda Mehrpouyan, Ion Madrazo Azpiazu, and Maria Soledad Pera. 2017. Measuring Personality for Automatic Elicitation of Privacy Preferences. In *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, 84–95.

[31] Anders Møller. 2017. dk.brics.automaton – Finite-State Automata and Regular Expressions for Java. `http://www.brics.dk/automaton/`.

[32] James H Moor. 1997. Towards a theory of privacy in the information age. *ACM SIGCAS Computers and Society* 27, 3 (1997), 27–32.

[33] Tim Moses. 2005. Privacy policy profile of XACML v2. 0. *Oasis standard, OASIS* 2 (2005).

[34] Helen Nissenbaum. 2004. Privacy as contextual integrity. *Wash. L. Rev.* 79 (2004), 119.

[35] Minolini Nithyanandam. 2016. An active rule-based system for XACML 3.0. (2016).

[36] Raúl Pardo, Musard Balliu, and Gerardo Schneider. 2017. Formalising privacy policies in social networks. *Journal of Logical and Algebraic Methods in Programming* (2017).

[37] Raúl Pardo, César Sánchez, and Gerardo Schneider. 2018. Timed Epistemic Knowledge Bases for Social Networks. In *International Symposium on Formal Methods*. Springer, 185–202.

[38] Joseph Phelps, Glen Nowak, and Elizabeth Ferrell. 2000. Privacy concerns and consumer willingness to provide personal information. *Journal of Public Policy & Marketing* 19, 1 (2000), 27–41.

[39] Joseph Reagle and Lorrie Faith Cranor. 1999. The platform for privacy preferences. *Commun. ACM* 42, 2 (1999), 48–55.

[40] J Rose and C Kalapesi. 2012. Rethinking personal data: Strengthening trust. *BCG Perspectives* 16, 05 (2012), 2012.

[41] Herman T Tavani. 2007. Philosophical theories of privacy: Implications for an adequate online privacy policy. *Metaphilosophy* 38, 1 (2007), 1–22.

[42] Herman T Tavani and James H Moor. 2001. Privacy protection, control of information, and privacy-enhancing technologies. *ACM SIGCAS Computers and Society* 31, 1 (2001), 6–11.

[43] Que Nguyet Tran Thi and Tran Khanh Dang. 2012. X-STROWL: A generalized extension of XACML for context-aware spatio-temporal RBAC model with OWL. In *Digital Information Management (ICDIM), 2012 Seventh International Conference on*. IEEE, 253–258.

[44] Giuseppe A Veltri and Andriy Ivchenko. 2017. The impact of different forms of cognitive scarcity on online privacy disclosure. *Computers in Human Behavior* 73 (2017), 238–246.

[45] Samuel D Warren and Louis D Brandeis. 1890. The right to privacy. *Harvard law review* (1890), 193–220.

[46] Alan F Westin. 1968. Privacy and freedom. *Washington and Lee Law Review* 25, 1 (1968), 166.

On the Appropriate Pattern Frequentness Measure and Pattern Generation Mode – A Critical Review

Tongyuan Wang
tttyyyw@yahoo.com
TechEngine Plus Com
Montreal, QC, Canada

Bipin C. Desai
BipinC.Desai@concordia.ca
Dept. of Computer Science and Software Engineering,
Concordia University
Montreal, QC, Canada

ABSTRACT

The classic case pattern mining is a fundamental subject in data mining and big data science. The goal of the mining is to find correctly from a given dataset the patterns and their respective intrinsic frequentness. This paper examines two important yet misused instruments, the pattern frequentness measure "support" and the full enumeration pattern generation mode, which cause serious Overfitting thus deviate from the mining goal. A theoretic combined solution for the two critical issues is then proposed. This solution plus the equilibrium condition introduced in this paper forms a set of three fundamental rationality check criteria that every mining approach should observe. As such, the rationality of the mining theory and the reliability of the mining results would be substantially improved from the previous work. These together promise a significant change towards more effective pattern mining.

CCS CONCEPTS

• **Information systems** → *Data mining*; *Evaluation of retrieval results.*

KEYWORDS

data mining, frequentness measure, overfitting, pattern frequency, pattern mining, probability anomaly, selective pattern generation, underfitting

ACM Reference Format:
Tongyuan Wang and Bipin C. Desai. 2019. On the Appropriate Pattern Frequentness Measure and Pattern Generation Mode – A Critical Review. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3331076.3331125

1 INTRODUCTION

Starting with item-set mining [1] [5], pattern mining has expanded into many current data mining research areas, for instance, medical, genomic and so on. Extensive research to date has been reported on frequent pattern mining. Most of the research focuses on algorithms and computation performances, including scalability and memory optimization [19]. Performance is certainly important in dealing with large datasets, but the first important yet less studied issue is the theoretic establishment of the mining mechanism and reliability, including their definitions, measurements and test norms. Once the fundamental mining theory is established one could attempt the design, implementation and testing of the mining algorithms. This paper is a humble effort in this direction to clarify some basic concepts and issues embodied in the two important instruments used in the mining: the pattern frequentness measure and the pattern generation mode, such that the rationality of the mining theory and the reliability of the mining results can be improved.

Frequent pattern mining, as its name implies, can be simply defined as a "frequentness based mining". It differs from other data mining techniques, such as classification or clustering that use other characteristics of the datasets for the mining. Because frequentness is the only criterion, a proper measure of it is of fundamental significance. A more profound issue is what pattern a frequentness should be assigned to. For a given dataset, normally we do not know, *a priori*, the number and for each pattern its makeup. Conventional mining approaches use "support" to measure pattern frequentness, along with the use of the full enumeration mode to generate patterns. This measure and the mode, however, are not justified under our review based on established probability and statistics theories. The drawbacks and consequences of the use of the measure and mode are detrimental for an effective pattern mining. Our solution is a reformulated support combined with the selective pattern generation mode to get around the drawbacks.

The second section of this paper investigates and discusses the drawbacks of previous approaches. The third section discusses and proposes a solution for the shortcomings of the "support" measure. The fourth section reexamines the infeasibility of the conventional full enumeration pattern generation mode and identifies a selective mode to replace it. The fifth section analyzes and justifies the effectiveness of our proposed solution. The last section gives our conclusion.

2 RELATED WORK AND OPEN ISSUES

This paper focuses on pattern mining over non-continuous data sources. However, the principles discussed herein could be applicable to continuous data sources as well.

2.1 The problem and terminology

According to the application domains, pattern mining has developed from the early days' market-basket item-set mining to today's temporal pattern mining, spatial pattern mining, sequential pattern mining, health care data mining, genomic pattern mining, and so on.

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331125

Nevertheless, the fundamental problem, finding frequent patterns, is common in all of these tasks. As there remain unsolved important issues in pattern mining we start our investigation from the market mining problem. Historically, it is the starting mining model, and most subsequent mining subjects and techniques inherited the basic concepts and methods from the item-set mining. Meanwhile, most readers are familiar with the market dataset and it remains one of the top mined data types in recent surveys [2] [3]. However, we will see that what is discussed in this paper is generic and not restricted to the transactional (market) dataset mining.

Table 1: A Database (DBo)

TID	VID
T_1	V_1, V_4, V_7
T_2	V_2, V_4, V_7, V_8
T_3	V_2, V_6
T_4	V_1, V_6, V_8
T_5	$V_1, V_2, V_3, V_4, V_7, V_8$
T_6	V_5
T_7	V_4, V_7
T_8	V_5
T_9	V_1, V_2
T_{10}	V_1, V_2, V_3, V_8

Table 1 (DBo) is the dataset for the running example used in this paper. It can be seen as an abstraction of market transactions typically presented in published item-set mining articles [1] [5] [19]. The DBo has u rows and two columns. Column TID represents the key attribute and VID represents an application domain Ω of n distinct elements. Each row is a tuple, where T_i $(i = 1, 2, ..., u)$ is a tuple ID, and each cell of column VID contains a value V (or a set of values) of that domain. For example, in a market-basket problem, a TID could represent a transaction ID, and a value of VID, V_i $(i = 1, 2, ..., n)$, represents an "item" from the domain Ω of merchandise. Particularly, a combination of k distinct Vs is termed as a pattern $Z_k = (V_iV_{j...}V_s)$ of length k, or k-itemset in market-basket problem [1] [5]. In conventional mining approaches, the enumeration process of such combinations is called pattern generation. By convention, the number of occurrences of a pattern Z over the database is noted as S_z or $S(Z)$, named as the (absolute) *support* of Z. In some literatures, such occurrence is also measured as the (absolute) frequency of Z and noted as $F(Z)$. The relative support is a ratio and noted as s_z or $s(Z)$ [1] [19], such that:

$$s_z = s(Z) = count(Z)/|DBo| = F(Z)/u = S(Z)/u = S_z/u, \qquad (2.1)$$

where u = |DBo| is the total number of tuples, i.e., the cardinality u of DBo. For instance, from Table 1, we can get $S(V_1V_2) = 3$ and $s(V_1V_2) = 3/10 = 0.3$.

Obviously, (2.1) comes from classical frequency based probability concept, and s_z should be taken as the first link between probability theory and pattern mining. Particularly, since S_z or $S(Z)$ corresponds to $F(Z)$ as an absolute frequency measure, s_z or $s(Z)$ should be taken as the frequentness measure.

In statistics terminology, the dataset DBo is a sample of the real world application at hand. The cardinality u of DBo is the sample size; and a record (tuple) is a realized *event* of the sampling, hence a subset of Ω [16]. In data mining language, we call each original tuple (event) an *original pattern*, or an *original observation.* A TID can be taken as a sample label or trial ID, and the column VID refers to the set of *events* [6]. Based on these notations, the fundamental pattern mining problem can be stated as follows:

Problem 2.1: Given a dataset DBo as shown in Table 1 involving the universe Ω of n distinct elements of domain VID, output all patterns of the elements in any length, such that s_z of a pattern Z satisfies $s_z \geq s_{min}$, where s_{min} is a user predefined minimum support; such satisfactory patterns are *qualified patterns.*

The above is a general introduction to the basic concepts of item-set mining, but at this point we'd notice importantly that, although the dataset of Table 1 was originally initiated to solve the market mining problem [1] [5]. However, it can only be seen as an abstract dataset and could not fully reflect thus fully solve the market problem. While the basic reasons are given below, we will further clarify this issue in next sections.

1). From the literatures as summarized in the next subsection, the "item" presented in the datasets such as Table 1 assumes the same concept of the "element" in set theory. That is, an item or element is unique and atomic (indivisible) presented in each data tuple. We call the uniqueness and indivisibility properties together as the "classic data nature". This nature is reflected in the calculation of a single S_z shown above. Hereafter, if there is no ambiguity, we use the words "item" and "element" interchangeably.

2). Although each V_i in Table 1 is assumed to represent an item (product), the table is abstract. Particularly, in the very early itemset mining article [5], the mining problem is specified to deal with "a set of literals". Each element (item) is called a "literal", and coded with an ID as can be inferred from that article. This coding convention is adopted in later research papers especially the dataset reservoir [47], wherein all the elements stored in every dataset are coded and each element is represented with a unique number. That is, the semantics of each element in every dataset is ignored, such that all elements are taken to be homogeneous with only their numerical IDs left to distinguish each other.

3). The dataset is static.

With the above characteristics, we call the dataset such as Table 1 as the "classic dataset".

Since the classic dataset is abstract and could not exactly reflect the market problem, the so called itemset mining over it may not realize its mining purpose adequately. On the other hand, this dataset could closely model some other mining problems. For instance, if each tuple of Table 1 represents a daily record of a museum visition and each V_i represents a visitor to that museum, then V_i is naturally unique and atomic in a tuple. Considering these aspects together, we refer the classic dataset (Table 1) with Problem 2.1 together as the "classic case mining model" or simply the "classic mining problem". Accordingly, the mining issues and their solutions to be discussed in the rest part of this paper is generic and applicable to any mining problem of the similar nature of the classic mining but not restricted to item-set mining only.

The classic case model is the simplest thus fundamental mining model. Only after the mining theory and techniques on the simplest model have been well established, could we properly proceed to more complex mining applications. This is the basic significance we reexamine the classic case mining.

It is not too difficult to comprehend the classic mining problem, while the main issue for most data mining research is the computation complexity due to potentially the power set (2^n) of possible patterns over the n-element domain Ω. Note that, in this paper we do not take the empty set (Ø) as a pattern, and let its frequency $F(\text{Ø})$ as undefined (this is because every tuple could produce Ø, taking Ø as a pattern would lead to a couple of problems in pattern generation and frequency determinations). Hence the largest number of possible patterns is $2^n - 1$. The power set complexity demands not only a great amount of computation time but also large memory space to store the candidate patterns and other information. Whereas the computation complexity issue has been extensively studied, however, the more fundamental work on the appropriateness of frequentness measure (2.1) and the pattern generation mode seems to be neglected. In the next subsection we briefly review some of the work on pattern mining.

2.2 Related work

In the classic case mining, earlier mining approaches mainly focused on how to efficiently obtain all possible qualified patterns from a given dataset. Examples of these proposals are the *Apriori* algorithm [5] [19] and its variations and extensions, such as the "incremental mining" [34], the "dynamical item-set counting" [35], the "parallel and distributed mining" [38], the "hash-based" [39] and the "partitioning" [39] algorithms. The *Apriori* based approaches feature pruning infrequent item-set as early as possible to achieve computation efficiency by use of the downward closure property which states that the super patterns of less frequent patterns could not be frequent [5] [19]. Subsequently, an approach called frequent pattern growth (FP-growth), originally developed in [7], tries to achieve efficiency by avoiding candidate pattern generation so as to reduce memory space requirement and IO cost. The avoidance is achieved by a construction of frequent pattern tree, or FP-tree, which is a prefix tree and acts as a compression of the original database, such that a data tuple is embodied in a branch of the tree. Then pattern mining from the original database is converted to mining from the FP-tree. This approach has also been further developed with many extensions and variations, such as the "hyper structure mining" approach [21], the "bottom up and top down" tree building approach [41] [32], the array based data structure to implement the prefix tree [42], and the like. There is also a proposal which avoids multiple IOs and reduces the candidate pattern generations by statistical estimation based on database scanning [8]. Other approaches try to improve mining efficiency by different pattern search strategies, including "breadth first" and "depth first" search methods. In a breadth first search the patterns are generated and examined from each record of (T_i: {V_j}) in horizontal data format [19]. In a depth first search, the original dataset DBo is transformed into a "vertical" dataset (V_j: {T_k}) before patterns and their frequencies are determined [9] [10].

The above approaches and many others not listed have a common feature that all possible qualified patterns are produced and collected in the mining result set and delivered to the user. The result set is usually very large for a fairly large dataset, leading to the problem of interpreting the result. Researchers tackled this issue by presenting the mining result set in a reduced form.

The "constrained" pattern mining [20] [21] [22] reduces the mining result set size by user constraints. For instance, a user may want to mine patterns with V_1 and V_2 only from Table 1. Another school of reduction approaches is the "concise" ("condensed" [4], or "compressed" [30] [31]) representation of the patterns, where a small subset of the frequent patterns is used to represent the whole mining result set. For example, the "free sets" [24] or "generators" [25] are concise sets to represent the whole result set in an application. Similarly, other concise sets and mining approaches, e.g., "disjunction-free" [26], or "non-derivable" sets [23], the "succinct summarization" [29] and the "krimp" [33] approaches, have been proposed, while the "closed" [11] [13] and the "maximal" [14] [15] [**?**] approaches have attracted more attentions. A pattern is closed if none of its proper super-pattern takes the same frequency [13] [19]. A pattern is maximal if none of its proper super-pattern is frequent against a s_{min} [13] [19] . The "closed set" representation is a lossless compression of the results set, in the sense that all patterns and their supports can be derived from the closed set; while the "maximal" expression is a lossy compression [19]. Similar to a lossy compression, there are approximation approaches to represent the mining results, for instance, the "top-k frequent patterns" [37] [48], "top-k most frequent closed" [27] [40], and the "pattern profile" [28] approaches.

Another big group of researchers propose to use the "interestingness measures" [54][55][58][57][62] or "quality measures" [63] called alternatively to reduce the mining result set. The interestingness measure school is complex and originated from dozens of reduction proposals not only for association rules but also for classification rules and summaries mining applications [57]. Yet there is no formal or generally accepted definition, the interestingness is taken to be determined by 9 criteria, such as conciseness, coverage, utility, and the like. These 9 criteria can be further categorized into three classes: objective, subjective and semantic based [57]. Particularly in pattern-association rules mining, there are in total of 38 objective interestingness measures including the original "support" and "confidence" measures, as summarized in [57] from the literature [56][59][60][61], let alone subjective and semantic based measures. It is thus not a trivial job to find a proper interestingness measure in an application [56].

Similarly, there are proposals to use weighted measures [64][65], for instance, by assigning different weights to the s_{min} for itemsets of different importance, to pursue a better mining.

To date a multitude of research papers related to pattern mining have been published. However, only a small portion of these could be cited in this part due to space constraint. And since our focus is on the classic case mining, for a full understanding of the problem and the history, we need to look into those early proposed concepts and approaches, Many later developed mining approaches such as those on unclassical datasets thus beyond our focus, e.g., stream data [71] [72], uncertain data [68][69] or data with different weights [66][67], etc., are not particularly mentioned hereto, but our discussion in this paper will also have important impact on these approaches. Since these approaches inherited the basic concepts and methods from the classic case mining approaches thus inherited their drawbacks stemming from the $s(Z)$ and the full enumeration pattern generation mode. After the theoretic foundation of the classic case mining has been reshaped and solidified, the remaining mining approaches will have to be rebuilt.

From the above summarized literatures, we see previous researchers have noticed the problem of excessive number of result patters in the classic case mining and tried many ways to remedy the problem. However, in our examination the problem is still not essentially solved, and the basic reason is that previous remedies are mainly on the squeeze of the size of the result pattern set by using different exogenous measures or constraints but not on the ill-definition of the measure $s(Z)$ and the infeasibility of the full enumeration pattern generation mode.

For instance, the interestingness measure or the weighted measure y_z over a pattern Z assumed by different researchers can be generally expressed as $y_z = f(s_z, X)$, where X is a vector of the proposed exogenous measures, modifiers or constraints by a researcher. Whether X is properly proposed or not, since the support measure s_z is not well defined (to be seen in the next section), y_z could never become sound thus could not function well.

Even though the concise or representative approaches are assumed to overcome the drawbacks of the full enumeration pattern generation mode so as to reduce the result pattern set size, they are based on that mode. For instance, for a given pattern Z, they produce the same s_z as other approaches. Secondly, the downward closure property is still kept in these approaches (this property is only a phenomenon of the full enumeration model as to be seen later). Thirdly, these approaches can not claim whether only patterns within the concise result set are meaningful while others outside the set are not. Additionally, the "closed", "maximal" and the like concepts are indeed other exogenous measures as well. These exogenous measures are created by the researchers (miners) but not from the users or without a clear connotation of the measures to the users. For instance, a user may just want the miner to deliver a set of reliable patterns to him but does not know what exactly the closed or maximal patterns mean to him.

We do not generally object to the use of exogenous measures or constraints in the mining, while our point is, only after the two basic instruments, the frequentness measure and the pattern generation mode have been well established could the supplemental measures be properly taken in and function effectively for the mining. This is the fundamental reason we need to restudy the issues of the two instruments as presented in this paper. We will see in the rest of this paper that by rationalizing the $s(Z)$ and using the selctive generation mode only, we will achieve a great reduction of the size of the result pattern set. Our approach is then not a concise or representative approach in conventional sense, and the reduction from our solution is thus fundamental, natural and unconditional. In contrast, previous reduction approaches are supplemental and conditional – by means of imposing constraints or extra measures, etc., to realize a reduction.

Since previous attempted remedies did not well solve the basic issues of the support and the generation mode satisfactorily, to save space, we won't discuss those remedies again in the next subsection but mainly the drawbacks presented in previous work.

2.3 The drawbacks of the previous approaches

As investigated in [50], from an observation point of view, the main drawbacks of conventional approaches exhibit in:

1) Meaningless but overwhelming number of resulting patterns, some being even "counter intuitive" [43].

2) The bias for generated patterns against the originally observed ones. For instance, from Table 1, $S(V_1) = 5$ while $S(V_5) = 2$, meaning V_1 is more frequent than V_5. However, V_5 is an originally observed pattern but V_1 is not surely a pattern yet, since it is "generated" thus only a "possible" pattern. Furthermore, in the next section, we will see, even by generation, $S(V_1)$ could not surely reaches 5.

3) The bias for shorter patterns against longer ones. This is formalized in the so called "downward closure" property [19] as mentioned before. Ultimately, the individual elements will be the most frequent patterns derived from the dataset. But in real world, it is common that compounds (patterns) are more frequently seen than single elements. For instance in a copper mine, pure copper (Cu) is much less frequently found than its compounds, e.g., copper oxide (CuO).

From a deeper analysis, the above phenomena or drawbacks come from the following main issues of previous work:

1) The improperly defined frequentness measure, the conventional "support". We discuss this and the related solution in the next section.

2) A lack of formal definition on what a pattern is, and the infeasible full enumeration pattern generation mode used conventionally.

In a sense, most of the above issues can be traced to the proper definition thus a well understanding of the pattern, although so many papers about pattern mining in general have being published so far. We are yet not ready to give it a formal definition here either, but at least, we can intuitively understand that:

A frequent pattern should be a configuration of the same elements appearing significantly in a dataset.

Configuration means combinations, but not only that. Configuration implies stability, structures, and more importantly internal inherent connections, known or unknown, among the elements of a pattern. In pure frequentness based pattern mining, we do not have to consider the structure issue, and we do not even have to know the reasons for internal connection before or during the mining. However, the nature of such connections should be addressed. Indeed, in many cases, revealing such internal connection reasons could be the main interests and purpose to undertake a pattern mining. When the nature of internal connection among the elements of a pattern is addressed, it is easy to see the inappropriateness of the full enumeration mode. Further discussion about this mode is presented in Section 4.

3 THE IMPROPER DEFINITION OF THE "SUPPORT" AND ITS SOLUTION

As stated before, pattern mining is a probability and statistics based technology, and some people even take pattern mining to belong to the domain of statistics [18]. Particularly, the frequentness measure s_z is the first link between pattern mining and the probability theory. However, under our investigation, a radical problem of pattern mining comes from the s_z as defined in (2.1). It can be easily observed that the accumulative probability (the sum of s(Z)s) from a mining project as presented in previous work in general is much larger than 1, which then seriously violates the fundamental probability concept. We call this issue a "probability anomaly".

A concrete example of the anomaly can be referred to Table 4 (in Section 5) based on the dataset of Table 1, where $\sum s_z > 11 \gg 1$.

In real applications this probability anomaly is much more severe, and the reason of it will become clear in the next sections. Such probability anomaly can only be traced to the improper definition of s_z (2.1). Indeed, we can formally prove (though the proof could not be presented here due to space constraint) and find the equivalence of s_z in both the classic frequency based probability theory and the multivariate probability theory [16][17], but neither of them justifies the use of s_z as the proper frequentness measure in pattern mining. For instance, s_z is equivalent to the absolute probability (or unconditional probability called alternatively) $P(Z)$ in the classic probability theory. Such $P(Z)$s, for instance $P(A)$ and $P(B)$, could not be used directly to compare the relative frequentness of A and B. Instead, $P(Z)$s are used in the conditional probability and dependence study. For example,

$$P(B|A) = P(AB)/P(A), \qquad (3.1)$$
$$P(D|C) = P(CD)/P(C),$$

where $P(A)$ and $P(C)$ are the absolute probability of A and C respectively. A numerical comparison between $P(B|A)$ and $P(D|C)$, or between P(AB) and P(CD) could not bring any semantic meaning, since they refer to different conditions. However, in pattern mining, s_zs are used for their direct comparisons in previous work.

In the above view we can see an extended issue that, the "confidence" measure [1], [5], [19] used in association rules mining of conventional mining approaches is an application of the dependence analysis, since the measure, for instance,

$$conf(A \rightarrow B) = s(AB)/s(A),$$

its right hand side is exactly the same as the right hand side of (3.1). That is, $conf(A \rightarrow B) = P(B|A)$.

However, it is not appropriate to compare the magnitudes between $conf(A \rightarrow B)$ and $conf(C \rightarrow D)$, for instance, simply because their counterparts $P(B|A)$ and $P(D|C)$ are not directly comparable as just addressed above. This is an example that the reestablishment of the pattern mining theory will unavoidably impact other related mining work. Yet, the "confidence" issue is not to be discussed further here, since it is beyond the focus of this paper and because of the space limitation.

A more understandable yet not well noticed point is that in the definition of $s_z = S_z$/u, where u is the cardinality of the original dataset DBo, but the collection of the result patterns Zs should be seen as a new dataset out of DBo. We name such datasset as DBv, and suppose each pattern Z to be stored in a separate tuple of the DBv. Then the accumulative frequency (occurrence) of all the result patterns equals the cardinality of the DBv. That is, s_z should be reformulated as:

$$s'_z = S_z/w, \qquad (3.2)$$

where, S_z represents the generated "occurrences" or "frequencies" of Z, $F(Z)$, while w is the cardinality of DBv. That is, $w = |DBv| = \sum S_z = \sum F(Z)$.

With the above reformulation, $\sum s'_z$ will automatically become 1, and the probability anomaly is fully eliminated. Meanwhile, we can easily notice that $s'_z \leq s_z$ will hold in general, while for most s'_zs, $s'_z \ll s_z$ will be true. This will become clearer in Section 5.

We notice that there might be arguments against the above observations and the reformulation. A most possible one is that the "supports" $s(Z)$s should not be directly summed up together, since, for instance, for a given pair of patterns A and B, they may be conjunct such that $s(A)$ and $s(B)$ cannot be directly added up. We appreciate that this argument implies an acceptance of the use of the probability theory to look into the problem. On the other hand, we notice that in the next section we will clear up such conjunction issue such that all $s(Z)$s can be directly additive.

Another argument might be why should we uniform the sum of the supports into 1? We can answer this from different aspects, while a direct reflection is that this argument implies a refusal of the probability theory to view the $s(Z)$s. If so, one should give his/her reason so, and specify on what theory the support measure is established. In fact, for instance, when one took X as a frequent pattern with $s(X) \geq 20\%$ in an application, s/he indeed compared that 20% with the reference number 1. Otherwise such a 20% could be trivial compared with a reference number much larger than 1.

Additionally, notice that the absolute support S_z is no better a choice than the relative s_z.

The above reveals that use of $s(Z)$ is a major cause of too many frequent patterns exhibiting in a mining application, the first drawback of previous approaches (subsection 2.3). This is because most $s(Z)$s are larger than their real frequentness as implied in previous paragraphs. Theoretically, such an inflated valuation of the pattern frequentness is referred as an overfitting matter. The formal concept and measurement of "overfitting" is originally from the theory of numerical statistics modeling, which however could not be directly adopted in pattern mining. A simple understanding here is that, overfitting means an over evaluation of a pattern frequentness, such that a spurious pattern is falsely taken to be significant one. Conversely, a true frequent pattern but falsely taken to be infrequent is termed as underfitting, due to an under evaluation of the pattern frequentness. In our study, the overfitting problem is widely presenting in previously work, as to be seen further in the following sections.

The overfitting or underfitting problem is important since it determines the reliability of the mined results. Reliability is a widely used and discussed criterion in data mining community. Article [49] is an example wherein the problem of enhancing data mining reliability is addressed. However, formal and concise definition of data mining reliability and its measure is not readily available. In general, we would take the concepts used in classic statistical tests as a reference to express the mining reliability. These would include the stability of the mined results against data size change or data source change, and more importantly the degree of closeness of the mined results to the real values or structures embodied in the real world. For an unknown world, the said closeness can only depend on the soundness of the mining technology. The minimum requirement of the soundness should be the compliance of mining results with commonsense or already known facts; a higher requirement should be the conformability of the mining principles with other related established theories.

The above analysis reveals that the conventional "support" is inconsistent with the basic concepts of the established probability theories thus needs to be reformulated and their summation must be equal to 1. What emphasized here is the importance of the establishment of a rational underlying measurement system for the mining, since, no science or technology can be recognized as well-established without a well-established measurement system. For instance, the first part of a typical textbook of physics [70] is on its measurement system.

4 THE APPROPRIATE PATTERN GENERATION MODE, SELECTIVE VS. FULL ENUMERATION

The previous section explains why $s(Z)$ is not the proper pattern frequentness measure and should be replaced with $s'(Z)$, such that the probability anomaly will be automatically eliminated and the overfitting be greatly reduced. However, $s'(Z)$ alone could not fully correct the overfitting or underfitting without a proper pattern generation mode. In the following, we examine the problems related to the full enumeration mode used by the conventional mining approaches and propose a selective generation mode to replace it.

To proceed, we may want to see why pattern generation is performed. Assume V_2, V_4, V_7 and V_8 represent egg, coffee, milk, and flour respectively in a market transaction T_2 of Table 1, and the customer who made this transaction indeed wanted to combine coffee with milk as one pattern, and use egg with flour to make cake as another pattern. Then T_2 truly contains at least two patterns. The purpose and significance of pattern generation is then to recover those patterns implied in a tuple. Based on this interpretation pattern generation is justifiable. However, the full enumeration generation over every tuple is not justifiable. This is because there will be three cases when a tuple is collected:

a). The entire tuple is a single pattern.

b). A coincidence for these elements to come together, i.e., the tuple contains no real pattern but a random walk.

c). The tuple contains more than one pattern.

We see only in the third case could a pattern generation be needed. Note that, similar to most of the previous work, for simplicity this paper ignores the second case and assumes at least one pattern would be produced from each tuple. What to be addressed here is, even in the third case, the full enumeration generation mode is still not generally applicable. To see this, we analyze why this mode is used by previous mining approaches. There might be two causes:

The first cause could be the confusion between pattern formation in the real world and pattern generation from the abstracted classic dataset. Since the mining research was started from the market problem, in a real market transaction an item is usually in plural quantity or divisible or both. For instance, the item egg would represent a box (or boxes) of 6 or 12 eggs, each of which can be used for different purposes and times. Similarly, the item milk can be divided and used for different purposes and times. Full enumeration mode could be possible then. However, in the classic case mining, the dataset is static and the plurality and divisibility properties of the items got lost, leaving each item in a tuple being unique and atomic (the classic data nature). The static and classic natures make the full enumeration pattern generation mode impossible, since these natures disallow the reuse of any item to generate different patterns within the same tuple. This observation also reflects the nature of mining: a mining means to reveal and identify whatever patterns exist at the mining time only but not to be concerned with when such patterns had formed or to reshape later. In the above example, after V_7 had been used with V_4 to form a pattern, V_7 is consumed and could not be used again to generate other pattern(s) from that tuple in a mining.

Notice that the classic data nature is reflected in the calculation of tuple length and pattern length. We can also see the reflection in the calculation of a single $S(Z)$ in conventional approaches since $S(Z)$ is incremented by 1 only for each occurrence of Z in a tuple (this may be easier to see when Z is a length-1 pattern, a single item), but collectively the classic data nature is violated since an item can be found in more than one pattern generated from a tuple.

As we all know, for any theory, once its initial model including the concerned concepts, assumptions, hypotheses and so on has been established, the rest part of the theory must maintain those establishments, otherwise the theory will fall into self-contradiction and can never become sound. The above addressed issues exhibit the contradiction and unsoundness of the previous mining approaches.

Table 2

TID	VID
T_1	A, B
T_1	B, C
T_3	A, B, C

At this point, one may ask why not consider more properties of the items in the mining model such that the above discussed issues be overcome and we achieve a real world mining? The answer is simple: doing so will change the mining problem, that is, the mining methods and mining results will be substantially changed from the conventional ones, at the same time the mining complexity will be greatly increased to an unmanageable level. To see this clearly and for simplicity, hereunder we use a mini classic dataset of three elements and three tuples (Table 2), and add in the plurality factor only into it to form a new dataset (Table 3), to demonstrate the changes and the mining complexity.

Table 3

TID	VID
T_1	A(2), B(5)
T_1	B(1), C(7)
T_3	A(3), B(3), C(5)

(1) The change of the design of the data structure and/or the database scheme to hold the added properties, as seen from Table 2 to Table 3, where the parenthesized number in Table 3 means the quantity (units) of the related item.

(2) The change of definition and calculation of $S(Z)$ and s_z. Conventionally from Table 2, $S(B)$ = 3 and $S(C)$ = 2, meaning B is more frequent than C, but from Table 3 the conclusion is reversed, since $S(B)$ = 9 and $S(C)$ = 12. This is a natural outcome, since quantity is taken into account, then the quantity matters.

(3) A more serious change is on the pattern expression (formula) and the great increase of the number of patterns. From Table 3, we will have patterns like A_2B_3, or B_3C_4, or $A_rB_sC_t$ in general, where the subscripts are integers each representing the quantity of the respected item within a pattern. We call such a pattern the "complex pattern" or "general pattern". Note in the classic case mining, only the simple form patterns such as *AB* or *ABC* are concerned, because the quantity factor is ignored there. To our best knowledge, no article has touched the concept of complex pattern in the itemset mining reaserch. Obviously, the complex patterns are much more generally presenting in the real world than the simple patterns,

and the mining complexity of the complex patterns will certainly become unmanageable by any mining approach proposed so far. This is because the number of possible complex patterns formed from a fairly large number of items will be astronomical, if not infinite. Notice the complexity of the power set in the classic case mining is already hard to handle.

Readers would have seen the complications now, especially when the divisibility and other factors were also taken into the mining model. The inclusion of the divisibility into the model will not only change the problem again, but also induce another difficulty to the researchers: what rules and how to set up the rules to render the division for different items!

The above reflected what we stated before: we are still far away from the goal of real world pattern mining, since we even have not well pursued the simplest classic case mining yet, though so many mining research papers have been published so far.

The second cause of the use of the full enumeration pattern generation mode by the conventional approaches could be the neglect of the difference between the "possible" patterns and the "realized" or "deliverable" patterns. A strong proof of this is that most early proposed approaches as summarized in Section 2 aim to output "all possible patterns" of $s_z \geq s_{min}$ (notice s_{min} can be zero), and the words "realizable patterns" or the like are seldom seen from those early papers, nor from later developed concise or representative approaches. That is, even if previous researchers had noticed the problem of too many result patterns, they did not well understand what the real reason behind is. Particularly, they did not consciously notice that the patterns to be delivered to the user must be firstly realizable from the mining, they thus could not propose effective approach to solve the noticed problem. Obviously, the realizable patterns will be much fewer than the possible ones. This can be seen from the example tuple $\{V_2, V_4, V_7, V_8\}$ again, from which V_4, V_2V_4, V_4V_7 and V_4V_8 are possible patterns, but at most one of them can be realizable, because of the uniqueness of V_4 as mentioned before.

The realization concept is another key to see the inappropriateness of the definition and calculation of the support $S(Z)$. If the original idea of the $S(Z)$ is to mean how many tuples of a dataset to support a given pattern Z, it is fine for a single Z but collectively it is not. In the above example, if V_4V_7 is taken supported by the given tuple, then V_4V_8 or so could not be. In general, we are unable to determine the $S(Z)$s since we do not know how many and what patterns could be supported by a particular tuple before a sophisticated theory has been established, but conventional approaches take this as an easy job by assuming each tuple could support all the possible patterns from the elements it holds, as implied by the full enumeration mode.

The analysis of the above two causes would have helped us understand why the full enumeration pattern generation mode is not feasible in the classic case mining. Notice additionally, even in real life the full enumeration mode cannot be arbitrarily assumed since real pattern formation is attribute constrained. For instance, to form coffee and soap into a pattern does not make much sense. However, since the classic dataset is de-semantic, as in most previous research, such constraint is ignored herein. We now focus on the feasible pattern generation mode and its properties over the classic datasets.

Since pattern is generated from each tuple, we take tuple T_1 of Table 1 to start the discussion. The tuple holds elements V_1, V_4, and V_7. Theoretically $7(= 2^3 - 1)$ "possible" patterns can be generated from it, but practically at most three out of the seven patterns can be "realized" and delivered in a particular mining operation. This is because, in an application, only one of the following optional pattern sets can be realized (delivered) from the tuple:

Example 4.1: The pattern set **delivery options** of the given 3 elements:

a) $\{V_1V_4V_7\}$;
b) $\{V_1, V_4, V_7\}$;
c) $\{V_1, V_4V_7\}$;
d) $\{V_4, V_1V_7\}$;
e) $\{V_7, V_1V_4\}$.

Note that the "delivery" mentioned above and hereafter refers to a full set delivery without a consideration of the effect of s_{min} unless otherwise specified.

From the above option list we can see that, the first two pattern sets are **trivial**: either a) all elements assemble together as a single pattern, or b) each element stands separately as a pattern. For any nontrivial pattern set from c) to e) to be outputted and delivered in an application, the number of patterns is only 2, less than the number of the elements, but conventional mining approaches take all the 7 patterns realizable and deliverable, since all the 7 patterns will be outputted and the frequency of each of them will be incremented by 1 in the result pattern set.

The above observation can be extended to a set of any number of elements, and we have the following:

Theorem 4.1 (Pattern set size theorem). *The number of nontrivial patterns to be generated and delivered from a given data tuple in a particular application is less than linear to the tuple size.*

Proof. For a given set of finite k ($k > 1$) distinct elements, generating patterns from it equals to partitioning the elements into different groups, thus each element can be in one group only. It is then obvious to see that, the largest possible number of groups is k, wherein each element stands alone as a group (a pattern). However, such a partition and the resulting pattern set are trivial. Therefore, in any nontrivial output set the number of result patterns can only be less than k. □

Note importantly in the above proof, the pattern generation indeed can be seen as a partition of the original element set. Formally, A "partition" $\{ H_i \}$ of a given nonempty universe Ω means that [52][53]:

$$\bigcup_i H_i = \Omega, \text{ and } H_j \bigcap H_k = ,$$

where $|\{H_i\}| = m > 0; 1 \leq j, k \leq m,$ and $j \neq k$; each H_i is nonempty and called a "part" or "block" of the universe, or a "hypothesis" of the partition.

We define the partition based generation as the **"selective pattern generation mode"** (or in short, "selective mode"), meaning an element may or may not be drawn to form a pattern with other element(s) of the same tuple, depending on what partition to choose. For instance, each delivery option listed in Example 4.1 corresponds to a particular partition of those given three elements. The mining approach based on this selective mode is called "selective pattern generation based mining approach", or simply "selective approach".

We notice that only this selective mode is feasible, while the conventional full enumeration based generation mode is infeasible. To see this, we introduce the following fundamental criterion first.

Similar to the well-known "chemical equilibrium equation", we can have an **"input-output equilibrium function"** (or simply "equilibrium condition") for pattern generation as stated below:

Proposition 4.1 (Equilibrium condition). *The total count $C(e_i)$ of each element e_i embodied in the resulted pattern set must be equal to (or no more than) the total count $S(e_i)$ of the same element from the original element set (e.g. a tuple). That is:*

$$C(e_i) \leq S(e_i). \tag{4.1}$$

The "less-than" operator expressed in (4.1) is applicable to the case that some element(s) of the original element set are random walk and could not be formed or included in a pattern.

Example 4.2: Any of the delivery options of Example 4.1 satisfies the above equilibrium function. For instance, the last option e), which means:

The original elements: $\{V_1, V_4, V_7\}$ = The elements of output: $\{V_7, V_1V_4\}$.

Consider the element V_7, from the left hand side of the above, $S(V_7) = 1$, and from the right hand side $C(V_7) = 1$, such that $C(V_7) = S(V_7)$ holds.

However, in the conventional full enumeration mode, the above equilibrium condition does not hold:

$\{V_1, V_4, V_7\} \rightarrow \{V_1, V_4, V_7, V_1V_4, V_1V_7, V_4V_7, V_1V_4V_7\}$.

And consider the element V_7 again, from the left hand side of the above, $S(V_7) = 1$, but from the right hand side $C(V_7) = 4 > S(V_7)$, violating (4.1)! Furthermore, in a real mining application, we will get $C(e) \gg S(e)$ as can be imagined and the equilibrium condition will be seriously violated.

The above violation can be interpreted in other way: originally, only single V_1, V_4 and V_7 are given, but in the conventional approach and the 7 patterns produced, it means there should have been 4 V_1s, 4 V_4s, and so on to form or to "support" the 7 patterns. It is certainly a lossy business if a miner received the original 3 elements but turned the 7 patterns back to the user according to the full enumeration mode or any other mode violating the equilibrium condition (4.1). However, interestingly conventional mining work all take this lossy approach.

Corollary 4.1 (Equilibrium condition-2). *For a given dataset or any number of tuples of it, let the total number of all elements forming the result patterns $C_t = \sum_i C(e_i)$, and the total number of original elements from the tuples $S_t = \sum_i S(e_i)$, then*

$$C_t \leq S_t. \tag{4.2}$$

The semantics of the above corollary is obvious: what outputted could not be more than the inputted. The equilibrium condition implies both (4.1) and(4.2), where (4.1) means an elemental view while (4.2) as the summation of the both sides of (4.1) represents an aggregative view of the condition (as such, (4.1) implies (4.2) but not vice versa). Particularly, for a single tuple t, $S_t = b_t$, the length of that tuple, and $C_t = \sum |Z^i|$, the sum of the lengths of the patterns $\{Z^i\}$ generated from that tuple (i is a cardinal number). For an entire dataset,

$$\begin{aligned} S_t &= \sum_{t=1}^{t=u} b_t, \quad and \\ C_t &= \sum (|Z^i| * F(Z^i)) \\ &= \sum_k (k * \sum_i F(Z^i_k)) = \sum_k (k * H^s_k), \end{aligned}$$

where Z^i_k is any pattern of length k from the entire pattern set, and $H^s_k = \sum_i F(Z^i_k)$ is the sub-accumulative frequency of all the patterns of length k from the entire dataset.

Since the equilibrium condition is very fundamental, it can be very useful in applications, especially in checking the correctness of a pattern generation from either a single tuple or a group of tuples or an entire dataset. One can easily take the running example (Table 1 to see how the conventional approaches violate this equilibrium condition.

Furthermore, the equilibrium condition is also applicable to the case that plurality (repetitions) and/or divisibility of any object in a mining needs to be considered. In this case, we only need to change the word "count" (meant by $C(e_i)$ and $S(e_i)$) into "quantity" or "volume", or the like, in an application.

More importantly, the equilibrium condition can be used to justify the proposed "selective pattern generation mode" and Theorem 4.1, as seen below.

Theorem 4.2. *The result patterns from a tuple of the classic dataset can only be a partition of that tuple.*

Proof. Due to the classic data nature and based on the equilibrium condition, no element can belong to more than one result pattern from the same tuple. As such, the result patterns from a tuple can only be pair-wise exclusive and no super and sub-pattern relation could exist within a tuple. This is the same to see that the result patterns from a tuple can only be a partition of that tuple (refer to the formal definition of the partition on the upper part of the left column). □

The above theorem is equal to saying that only the partition based selective pattern generation mode is feasible.

Theorem 4.3. *For a given data tuple, in the classic case and the selective pattern generation mode, Theorem 4.1 and the equilibrium condition are mutually necessary and sufficient.*

Proof. To save space, we prove only one part of the above theorem: In the classic case Theorem 4.1 holds *iff* the equilibrium condition holds under the selective generation mode.

(1) The necessary condition can be proved by contradiction: Assume Theorem 4.1 hold but (4.1) do not, i.e., there exist at least one element x such that $C(x) > S(x)$, which means more than one pattern would contain x, or a pattern would contain more than one x. However, such pattern(s) could not be produced from the selective pattern generation mode by its definition in the classic case mining, and Theorem 4.1 could not hold thus a contradiction of the assumption.

(2) The sufficient condition can also be proved by contradiction: Assume the equilibrium condition hold but Theorem 4.1 do not, i.e., (ignore the trivial case) the number of generated patterns be

more than the number of the original elements. Since a pattern contains at least one element (as noted before), the total number of elements to form the patterns must be more than the number of the original elements, such that $C_t > S_t$ happens, which violates the equilibrium condition (4.2) and contradictory to what assumed. □

The above theorem means, once the equilibrium condition holds, the pattern set size of Theorem 4.1 would automatically be satisfied.

The concept of the selective mode is simple, although how to render this mode would be discussed in a future paper. This simple concept based mode will have important effects and significances for proper pattern mining. The most significant one is the theoretical removal of the concern of conjunction issue in the additivity of the pattern frequentness, as shown below:

Theorem 4.4 (disjuction property theorem). *The patterns produced from the selective generation mode are conjunction issue free.*

Proof. For a single tuple, as stated in the proof of Theorem 4.2, patterns produced from this mode is pairwise-exclusive thus conjunction free. This can be verified by any delivery option of Example 4.1.

For the whole pattern set produced from an entire dataset, there could be some patterns containing the same element(s), for instance V_1V_2, named as A for simplicity, and V_2V_3, named as B, from Table 1, but they can only be produced from different tuples because of the above reason. This means there will be zero instance of their co-occurrence from the entire dataset, thus the probability $P(AB) = 0$, which is of the same effect of exclusivity of two patterns. □

Due to the disjunction property of the patterns produced from the selective mode, and based on the probability theory [16][17], the direct additivity of the pattern frequentness and the remedy to the probability anomaly stated previously are now fully closed with theoretic proofs.

An important notice here is that, although within the same tuple patterns are pairwise exclusive thus no super-pattern and sub-pattern relation exists as stated in the proofs of Theorem 4,2 and 4.4, such relations do exist in the entire dataset. The only thing is that those super and sub patterns come from different tuples as implied in the second part of the above proof. This is another major point that was not clearified in previous approaches

In addition to the above theoretical contributions, the practical important effect of our solution through the selective mode is the large reduction of the number of resulting patterns from the power set to less than linear (by Theorem 4.1), especially when the number of elements of a tuple is large. For instance, take a data tuple length of 40, not a very big number yet compared with that can be found in many data sources, e.g., in the mentioned dataset reservoir [47], by conventional approaches, more than a trillion patterns will be generated from the tuple and each pattern's frequency will be incremented by 1, but in the selective mode and from Theorem 4.1, at most 40 patterns could be rationally produced so. This is a striking difference and an evident approach to understand why too many (meaningless) patterns are usually produced from the conventional mining approaches.

Another important effect is the change of the frequencies thus the frequentness of the patterns from that of the previous work. The changes mean a reordering of the frequentness of the patterns. This is obvious since the number of times a pattern to be produced from the data tuples will be changed from the full enumeration to the selective mode. And notice that the relative order of a pattern aginst other patterns is more important than the frequentness number itself, since the frequentness numbers serve for ordering.

As an illustration of what stated in the above two paragraphs, suppose in an application, by conventional mining approaches the ordered result set be $\{A, E, D, B, F, G, C\}$, while in the selective mining from the same dataset, the ordered result set be $\{D, F, A\}$. That is, both the number of patterns and the frequentness order of the patterns changed.

The above means that the mining results from the proposed solution will be substantially different from that of the conventional mining approaches. Those approaches differ from each other mainly in algorithms but they all produce the same result pattern set, other thing being equal.

Detailed analyses of the effects of the selective mode together with the use of the reformulated support s'_z are given in the next section.

5 EFFECT ANALYSES OF THE PROPOSED SOLUTION

The reformulated pattern frequentness measure s'_z combined with the selective pattern generation mode proposed in the previous sections forms the fundamental solution for the issues addressed in Section 2 and 3, especially the central overfitting problem. As a summary, the main functionality and advantage of the selective generation mode is in the reduction of the meaningless patterns, but this mode must be used with the new measure s'_z together otherwise the overfitting problem would still be retained if s_z is used. The main functionality and advantage of the new measure s'_z is in the reduction of the overfitting because of its remedy to the probability anomaly. However, if only the s_z is replaced but the full enumeration mode is still in use, the following problems exist: the conjunction issue will remain and the direct additivity of the pattern frequentness hence the remedy to the probability anomaly would still be questionable. Secondly, s'_z, alone could not fully eliminate the overfitting, and it may cause underfitting at the same time. Thirdly, the order of the pattern frequentness would not change. The details are given below, and the conclusion is that the combined solution must be used either theoretically or practically.

To compare s_z and s'_z, we need to get the accumulative pattern frequency w. In the rest of this paper, w will be used for either selective or full enumeration pattern generation mode if there is no ambiguity, while w_0 will be used as a special value of w particularly for the full enumeration mode. We note that w_0 can be obtained without rendering the pattern generation as shown in the next subsection. We present the derivation of w_0 since it may be used to compare with a w from a particular selective generation approach.

5.1 The computation of w_0

The accumulative raw frequency w_0 of all possible patterns under the full enumeration mode can be obtained precisely before the pattern generation as follows:

$$w_0 = \sum_{j=1}^{j=u} \sum_{i=1}^{i=b_j} C_{b_j}^{i} = \sum_{j=1}^{j=u} (2^{b_j} - 1) = \sum_{j=1}^{j=u} 2^{b_j} - u, \tag{5.1}$$

where $b_j = |T_j|$, the number of elements in a tuple T_j, and $u = |\mathrm{DBo}|$.

Now, define g_k as the number of tuples each holding k elements in the original dataset DBo, such that

$$\sum_{k=1}^{k=\alpha} g_k = u, \tag{5.2}$$

where $\alpha = max(b_j)$.

(5.1) then can be further simplified as:

$$\begin{aligned} w_0 &= \sum_{j=1}^{j=u} \sum_{i=1}^{i=b_j} C_{b_j}^{i} = \sum_{k=1}^{k=\alpha} (g_k \sum_{i=1}^{i=k} C_{k}^{i}) \\ &= \sum_{k=1}^{k=a} g_k (2^k - 1) = \sum_{k=1}^{k=a} g_k \, \lambda_k \end{aligned} \tag{5.3}$$

where,

$$\lambda_k = 2^k - 1,$$

which represents the number of all possible patterns and hence the sum of their incremented frequencies enumerated from a tuple of length k.

The simplification of (5.1) to (5.3) reduces the number of exponent operations from u to α: commonly $\alpha \ll u$ in real applications. Furthermore, the exponent operations can be completely avoided, and w_0 can be calculated in a recursive approach (not to present). We note that the exponent operation is not a big issue in terms of computation cost. However, the computation cost of the addition operations of (5.1) is more than linear to u thus could not be ignored when u is very large. For instance, if u is in trillions or even larger, then computation of (5.1) may take hours or days with a current desktop system. However, the computation cost of (5.3) can be seen as near constant and negligible, since α is relatively very small and usually would not be over a hundred or a thousand in an application.

5.2 Overfitting/underfitting quantifications

In Section 3, we have shown how s_z defined in (2.1) is reshaped as s'_z in (3.2), and how the probability anomaly is primarily eliminated. Here, we present how the degree of overfitting or underfitting of conventional support s_z could be quantified against the reformulated one. We first define a primary overfitting or underfitting ratio r_s, depending on whether $r_s > 1$ or $r_s < 1$, for the quantification in the case that both s_z and s'_z are from the full enumeration pattern generation mode, and

$$r_s = s_z / s'_z. \tag{5.4}$$

We then get:

$$r_s = s_z / s'_z = (S_z/u)/(S_z/w_0) = w_0/u = \lambda_0, \tag{5.5}$$

where λ_0 is the average sum of frequencies of patterns generated per tuple of the DBo. That is:

$$\lambda_0 = \frac{w_0}{u} = \frac{1}{u} \sum_{k=1}^{k=u} \lambda_k = \frac{1}{u} \sum_{k=1}^{k=\alpha} g_k \lambda_k, \tag{5.6}$$

A merit of the λ_0 and r_s is that they can give us an immediate numerical magnitude of the overfitting ratio, since w_0 can be obtained from (5.3). Furthermore, as we notice that, λ_0 is increasing with the increase of the overall data tuple length, and so is r_s. It means then that the overfitting issue will be severer than that over datasets of shorter data tuples in conventional approaches.

Note that the overall data tuple length mentioned above is not exactly the average data tuple length with the full enumeration mode, but the dominant length of top long data tuples in a dataset. This is because: in this mode the number of patterns to be generated is exponential to the tuple length. We will touch this issue a bit further in the later subsections.

From the above we see that $r_s > 1$ always holds since $w_0 \gg u$ (if $w_0 = u$ it means no pattern generation), and r_s can be very large if the overall data tuple length is large. As an example indicated before, if the overall tuple length is 40, r_s will be in the range of trillions. It then justifies our previous assertion that overfitting is inherently embodied in previous mining approaches and strongly disqualify the extensively used conventional support s_z.

Note also that the overfitting ratio for the unnecessarily produced patterns in the conventional approaches will be theoretically extremely high (up to infinity ∞) since their frequencies S_zs will be zeroed in the selective mode. However, this infinity ratio is not reflected in r_s, since r_s is concerned with all the patterns produced in the selective mode only.

On the other hand, λ_0 hence r_s expressed in (5.6) is the upper bound of the overfitting ratio for real patterns with the conventional generation mode, thus the degree of overfitting for a real pattern would be lower than that from (5.6). This is because w_0 includes frequencies of those unnecessarily generated patterns from the full enumeration mode. To be clearer, if s_z is still used, the related overfitting ratio r_t is given below:

$$\begin{aligned} r_t &= s_z / s'_z |_{selective} \\ &= (S_z/u)/(S_z/w) = w/u = \lambda > 1, \end{aligned} \tag{5.7}$$

where $r_t > 1$ is due to $w > u$ because of pattern generation although $w < w_0$. Notice that, simpler than λ_0, λ can be approximately taken as the average tuple length of the entire dataset. This approximated λ indeed is the upper bound of λ based on the "pattern set size theorem" 4.1 with the selective pattern generation mode. This is because, the upper bound of w: $w_{up} = \sum_{j=1}^{j=u} b_j$, and $\lambda < \lambda_{up} = w_{up}/u$.

In comparison with λ_0 (5.7), normally $\lambda \ll \lambda_0$. This means the overfitting is much milder than that in the conventional approach even if s_z is used, and this is a direct benefit of the use of the selective mining approach. At the same time it proves what was stated above: the measurement of the overfitting will be lower down if the number of unnecessary patterns are less produced such that w_0 is reduced to w as done in the selective mode.

Table 4: Comparisons of the resulted parameters based on data of Table 1

k	#Z$_k$	$\sum$ s$_z$	$\sum$ s'$_z$	$\sum$s$_z$	$\sum$s'$_z$	Ex. Patterns	S$_z$	s$_z$	s'$_z$	S$_z$	s$_z$	s'$_z$
A	B	E	Ea	E'	Ea'	T	x	y	z	x'	y'	z'
1	8	2.56	0.33	2.8	0.24	V$_1$	4	.45	.039	5	.5	.042
2	18	3.33	0.36	3.6	0.31	V$_1$V$_2$	2	.22	.019	3	.3	.025
3	21	2.9	0.22	3.0	0.25	V$_2$V$_4$V$_7$	2	.22	.019	2	.2	.017
4	15	1.8	0.08	1.7	0.14	V$_1$V$_2$ V$_3$V$_8$	1	.11	.010	2	.2	.017
5	6	0.67	0.01	0.6	0.05	V$_1$V$_2$V$_3$V$_4$V$_7$	1	.11	.010	1	.1	.009
6	1	0.11	0.01	0.1	0.01	V$_1$V$_2$V$_3$V$_4$V$_7$ V$_8$	1	.11	.010	1	.1	.009
$\sum$	69	11.4	1.00	11.8	1.00							

Not only will the overvalued w_0 lead to overfitting, but also it induces undervalued s'_z. That is, s'_z is an overcorrection of s_z if w_0 is used, and s'_z will bring a underfitting problem to real patterns which keep the same S_zs in either mode. However, the degree of the underfitting will be smaller than that of the overfitting with the conventional support. Consider the underfitting ratio:

$$\begin{aligned} r_u &= s'_z|_{full-enumer.}/s'_z|_{selective} \\ &= (S_z/w_0)/(S_z/w) = w/w_0 < 1, \end{aligned} \tag{5.8}$$

The inverse of r_u should be used to mean the degree of underfitting, and the degree is thus defined to be:

$$r_u^o = 1/r_u = w_0/w > 1.$$

The following compares the degrees of underfitting and overfitting:

$r_u^o/r_s = (w_0/w)/(w_0/u) = u/w < 1.$

Since $u < w$ (otherwise no pattern generation), the above proves that the degree of underfitting is lower than that of overfitting for real patterns.

Note for patterns from the selective mode assigned with smaller frequencies than S_zs from the full enumeration mode, as can be inferred from (5.8), $r_u < 1$ does not generally hold. Accordingly, those patterns with $r_u > 1$ would still be overfitted in the full enumeration mode even if s'_z is used.

Now, since $w_0 \gg w > u$, the above implies the following relations:

$$r_s > r_u^o > r_t > 1. \tag{5.9}$$

That is, only by using the selective pattern generation mode, could the overfitting and underfitting both be minimized.

As a summary for the full enumeration based approaches, if the s_z (2.1) is used, overfitting exhibits persistently without underfitting, and the longer the overall data tuple length, the severer the overfitting. When s'_z (3.2) is used, real patterns will get underfitted, and the longer the overall data tuple length, the more serious the underfitting, while the degree of it will be much milder than the use of s_z. Lastly, in the full enumeration mode, even if s'_z is used and underfitting happens, overfitting is still unavoidable, because, at least the large number of unnecessarily generated patterns as stated before are overfitted ones by nature. In general, to overcome both underfitting and overfitting problems, the selective generation mode should be used.

5.3 Numerical comparisons between s_z and s'_z

For a more intuitive understanding of the difference of the evaluation of the pattern frequentness in conventional and the reformulated s_z, we present the related comparisons in Table 4 based on the data given in Table 1, and both s_z and s'_z are from the full enumeration mode. This is the only we can do presently, since, the implementation of the selective mode, thus the s'_z from this mode, can only be presented in a future work. Even so, the comparisons presented hereunder would still be informative.

In Table 4, column B is the subtotal number of patterns of the same length; column E shows the sum of s_zs of patterns of the same length, as well as the overfitting ratios (the last row) against the new s'_z, based on the first 9 tuples of Table 1. The semantics of column E' is the same as that of column E but based on all 10 tuples of Table 1. Column Ea and Ea' present the sum of s'_zs of patterns of the same length of the 9-tuple and 10-tuple cases respectively, where the probability anomaly is eliminated. The other columns starting from column T until the last show some example patterns and their related raw frequencies (x) and frequentness in terms of s_z and s'_z (column y and z respectively), where column x, y, and z are the results from the first 9 tuples, while column x', y', and z' are the results after the last tuple being added into Table 1.

As addressed before, overfitting is dominant in conventional approaches, exhibiting with two typical symptoms: too many frequent patterns and unstable mining result set. s'_z greatly remedy the first symptom. For this small example and under the conventional regime, the number of frequent patterns with s_{min} = 20% is 23 from the 9 tuples and 32 from the 10 tuples, a 40% increase for s_{min} = 20%, which illustrates that the result set is very unstable. The main cause of it is the probability anomaly which results in an overfitting ratio r_s > 11% in either 9 or 10-tuple case (the grand sum of s$_z$ in column E or E' means this ratio). More strikingly, at s_{min} = 10%, all of the 69 patterns are frequent from either 9 or 10 tuples in the conventional case!

Contrarily, in s'_z, because of the merits of its removal of the probability anomaly there is no frequent pattern even at s_{min} = 10% from the entire dataset. The result is compliant with an intuition that we could not mine a big number of frequent patterns from such a small element set and small database at a fairly high threshold s_{min} (e.g., at 10% or higher). Although as noticed in the previous subsection we need to consider the overcorrection, i.e., the underfitting effect of the s'_z in the full enumeration case, the effect is not much serious, since the overall data tuple length in this example is not large compared with that of real application datasets.

s'_z also (partially) remedies the second symptom, the unstable mining result set. This is because the decrease of s_z to s'_z reduces the number of patterns to pass the given threshold s_{min} to enter the result set when the data size is increased. The more complete remedy to the instability is the use of s'_z with the selective pattern generation mode together, since this mode reduces the number of patterns to be generated. Furthermore. with this mode, the reduction rate from s_z to s'_z will not be linear nor the same for different Zs.

Other features can be further inferred from the table: when using s'_z, the more frequent a pattern is, the more stable is its frequentness as the data size changes, as shown in column z and z' of Table 4. This is what is normally to be expected: with data size increasing, the frequentness of every pattern approaches asymptotically to its natural degree. In addition, s_z in general increases faster than s'_z. The theoretical reasons for these observations are given in the next subsection.

5.4 Theoretical comparisons between s_z and s'_z

The above observations can be formalized as a theorem as follows:

Theorem 5.1. *s'_z outperforms s_z in the following aspects:*

(1) *s'_z increases slower than s_z if $\Delta w > \lambda$, where Δw is the added accumulated frequency produced from the added data tuples, and $\lambda(= w/u)$ is the average accumulated pattern frequency per tuple.*
(2) *As long as the added data tuple contains Z, s_z always increase, while s'_z may not, or even decrease.*
(3) *A larger s'_z will increase slower than a smaller s'_z.*

In the following proof to the above, w and λ are generally used for either full or selective pattern generation mode, unless w_0 and λ_0 need to be specified.

Proof. Initial u, w, s_z and s'_z for a given pattern Z and its raw frequency F_z. Now suppose one data tuple added into the dataset, $\Delta u = 1$. It then could cause F_z to increase at most by 1, or $\Delta F_z = 1$, since one data tuple can generate a particular pattern once, but it could cause w to increase larger than 1, i.e., $\Delta w > 1$, since in general more than one pattern will be produced from a tuple, otherwise the problem becomes trivial. Then:

$$\begin{aligned}\Delta s_z/s_z &= \Delta(F_z/u)/(F_z/u)\\ &= (u\Delta F_z - F_z\Delta u)/u^2)/(F_z/u)\\ &= \Delta F_z/F_z - \Delta u/u = 1/F_z - 1/u, \end{aligned} \tag{5.10}$$

and

$$\begin{aligned}\Delta s'_z/s'_z &= \Delta(F_z/w)/(F_z/w)\\ &= ((w\Delta F_z - F_z\Delta w)/w^2)/(F_z/w)\\ &= \Delta F_z/F_z - \Delta w/w = 1/F_z - \Delta w/w. \end{aligned} \tag{5.11}$$

Since $w = \lambda u$, where λ is the average accumulated pattern frequency per tuple (refer to (5.9) and (5.6)), the above can be reformulated as:

$$\begin{aligned}\Delta s'_z/s'_z &= 1/F_z - \Delta w/\lambda u\\ &= 1/F_z - (1/u)(\Delta w/\lambda). \end{aligned} \tag{5.12}$$

(5.11) and (5.13) state that:

(1). $\Delta s'_z/s'_z < \Delta s_z/s_z$, as long as $\Delta w > \lambda$, which then proves the first conclusion of the thoerem. At the same time, it implies importantly, to keep $\Delta s_z/s_z$ comparable with $\Delta s'_z/s'_z$ when data size increases, that is, to slow down the rapid increase of s_z, the data tuple length will ultimately decline toward 1.

Note that, by full enumeration mode, Δw can be much larger than λ (or unambiguously, $\Delta w_0 \gg \lambda_0$) when a relatively long tuple is added in. Furthermore, since a bunch of patterns, including existent patterns, can be generated from the added tuple, it means the added tuple may cause a number of patterns' frequencies to be increased and ultimately get them to become frequent ones. Here the existent patterns are those generated before the addition of the new tuple. This explains how instability of mining result set and the overfitting problem take place at the same time in the full enumeration case. In the running example, λ_0 is about 12 for the first 9 tuples. When the 10th tuple (of 4 elements) is added in, $\Delta w_0 = 15 > \lambda_0$, and we have seen before how it causes a sudden increase of the number of frequent patterns thus an unstable mining result set, while now we get its theoretical reasons.

(2). (5.11) tells $\Delta s_z/s_z > 0$ always holds, since $F_z < u$ (if $F_z = u$, Z can be fully removed from the dataset, since every data tuple holds Z). However, (5.12) and (5.13) indicates that, even if an added tuple makes F_z increased (by 1), $\Delta s'_z/s'_z$ can be either positive or negative depending on whether the following condition hold or not as derived from (5.12) :

$$\Delta w * F_z < w. \tag{5.13}$$

If it holds, $\Delta s'_z/s'_z$ increases, otherwise decreases (ignore the equality case). It then proves the second conclusion of the theorem. This implies, similar but reverse to the overfitting issue of the above paragraph, under the s'_z regime and full enumeration mode, a fairly long added tuple thus a large Δw may cause a number of existent patterns' frequentness to decrease and hence underfitting happens, but the degree of the underfitting will be smaller than that of the above mentioned overfitting. We have proved this in subsection 5.2, and we can also prove it with the above formulas, but we do not have to do so.

On the other hand, we can see the above inequality (5.13) is easier to hold with a smaller F_z than larger one. This means another difference from the case of point (1) above that an added tuple can lead s_z to increasing only, but here it may cause some (less frequent) patterns' s'_z to increase while others' decrease. This is indeed a more proper reflection of the effect of data size changes, since such a change should alternate the comparisons of some patterns' frequentness. These two aspects together explains why the result set from the reformulated support s'_z is more stable than that from the conventional s_z

The above two points not only address the cause of overfitting and underfitting, but also reveal a data homogeneity issue. If the lengths of different tuples of the same dataset vary too much, then it may affect the proper measure of the mined patterns' frequentness hence the reliability of the mining. This is similar to the homogeneity problem in numerical statistic modeling: if the magnitudes of the data (numbers) differ too much in a sample, then it would be difficult to derive a reliable model from the data.

(3). Since a smaller F_z would let (5.13) easier hold, while a smaller F_z means a smaller s'_z, then (5.12) to (5.13) implies that a smaller s'_z increases faster than larger ones. It thus proves the third conclusion of the theorem as well. To be clearer, from (5.12), $\Delta s'_z/s'_z = 1/F_z - \Delta w/w$, for a given $\Delta w/w$, a smaller F_z thus smaller s'_z would give a larger value of the left hand side $\Delta s'_z/s'_z$.

Although on the above we added only one data tuple in the proof, the proof can be easily generalized to additions of multiple tuples. □

The above proof not only justifies the outperformance of s'_z over s_z, but more importantly reveals a number of interesting and intrinsic properties underlying pattern frequentness measure. For instance, the tuple length variation issue is seldom addressed by the previous work. Further more, the use of the a selective generation mode will reduce the effect of Δw hence the effect of the tuple length variations.

5.5 The significance and impacts of the proposed solution

The first significance is that we have found a solution for the problems identified in Section 2 and 3. The findings explored in our problem investigation and analysis indicate that, although pattern mining or "knowledge discovery from database (KDD)" in general is fact based (what a database holds are facts or experimental results), without a proper theory and mining strategy, we could not obtain reliable but misleading mining results. This paper and our solution is an attempt to clarify and correct some fundamental concepts so as to ultimately improve the mining reliability.

Another significance is the effectiveness of our solution. A well-established measure should be featured with at least rationality and simplicity. The simplicity implies easiness in understanding and in use. The proposed $s'(Z)$ keeps the simplicity while rationalizes the previous $s(Z)$ and remedies the probability anomaly. The use of $s'(Z)$ together with the selective mode could effectively get around the overfitting/underfitting and other issues addressed in Section 2 and 3. In particular, as indicated in the literature survey presented in Section 2, reducing the result pattern set size is a major research interest in previous work, and attempts have been made, e.g., [33] [48], but the reduction is not more than a few orders in magnitude, while with our new enumeration mode the number of patterns will be reduced from the power set to less than linear to the tuple size.

Furthermore, the equilibrium condition, the clearance of probability anomaly and the use of the selective pattern generation together forms a set of three rationality check criteria. Since these criteria are derived from the simplest classic case mining without involving any exogenous measure or constraint, every mining approach should comply with these criteria. In comparisons, readers would find that no previous approach did or could claim the satisfaction of these criteria.

Other consequences of the solution and criteria include:

1). Because of the equalization of the pattern frequentness and the probability measure of events, we can use 3% or 5% to be the frequentness threshold as used in various research and applications, though not formally required [45], such that there is no need to bother user (unless s/he likes) to define an s_{min}.

2) Under the s'_z regime and because of $\sum s'_z = 1$, in a case of large number of patterns, their frequentness distribution is more analogous to the (continuous) probability density distribution rather than the (discrete) mass probability distribution in probability theory. When using a 3% threshold for infrequent patterns under s'_z regime, it refers to all those patterns whose cumulated frequentness is less than 3%. Similarly, for the top 10% frequent patterns, it means their accumulated frequentness is no less than 10%.

3). The above impacts and consequences will unavoidably propagate to other mining applications based on pattern mining, e.g., association rules mining, causation mining, etc.

4). In practice, the more reliable mining approach and the greater reduction of the mining result set induced from the proposed solution would substantially simplify and facilitate business policy selection and decision making based on pattern mining.

5). The concepts and solution presented in this paper could also lay a foundation for more complicated mining applications, e.g., when plurality and/or divisibility issue needs to be considered in a mining, which we call the "plurality case" mining. In this case, an element represents a type of "objects", similar to that in chemistry study wherein an element represents each and all "atoms" of the same type. For instance, the element "O" represents each and all the oxygen atoms, and "C" for all carbon atoms, and so on. In the plurality case, complex patterns as mentioned in Section 4 will be generally produced, and we can follow the molecule formula approach to express such patterns.

For simplicity, the divisibility can be included in the "plurality case" mining as well, with an assumption that we could find a way to divide the concerned divisible element into its possible smallest parts each being treated as an atom upon a pattern generation, then the divisible elements can be taken as of the same plurality property as others. As such, the itemset mining could be more suitably studied in the plurality case mining.

With the above ideation, what established in the classic case mining could be easily extended into the plurality case mining. For instance, we have seen that the equilibrium condition established in Section 4 can be extended to the plurality case. Here we add that, the selective pattern generation mode and the pattern disjunction property theorem (Theorem 4.4) can be extended to the plurality case mining as well:

COROLLARY 5.1. *The partition based selective pattern generation mode is also the only feasible mode for the plurality case pattern mining, with the only notice that this mode is applied to the object (atom) level.*

The rationale of the above corollary is that, in the plurality case the smallest constituent unit of the pattern is the object (atom), not the type, and we thus need to look at the pattern composition in terms of atoms. In the atom level, each atom is unique and cannot belong to more than one pattern at a time. Then pattern generation from those atoms stored in a tuple can only be equal to a partition of those atoms, thus the selective mode and only this mode is applicable in the plurality case mining.

COROLLARY 5.2. *The pattern disjunction property is applicable to the plurality case pattern mining as well, since the patterns can only be produced from the selective generation mode.*

The above corollary would be easy to understand, since its proof will be the same as that of Theorem 4.4. As such, the direct additivity of the pattern frequentness can be applicable to the plurality case mining. As an example of the application, the corollary answers why in industrial mining practice, the percentages of the contents, CuO and FeO, etc., in a mining can be directly added up without a worry about the conjunction issue although the substances contain the same “O". The disjunction between CuO and FeO is because the oxygen atom involved in CuO is not the same oxygen atom involved in FeO when we look at these atoms individually, although they are labeled the same as “O".

As presented above, the concepts on the pattern expressions, the applicability of the equilibrium condition, the selective pattern generation mode, and the pattern disjunction property, they together would form the basic theory of the plurality case pattern mining. At the same time, these concepts are interdisciplinary conformable with other established theories, namely, probability, statistics, chemistry, and the list can be extended. This promises a realization of the application of pattern mining into computational chemistry [51].

6) We notice that this paper is only an initial part of the pursuance of the establishment of the fundamental pattern mining theory, which is a big work and could not be fulfilled with one paper only. For instance, more formal definitions and measures on overfitting and underfitting may be desired; the reliability theory and measures on mining methods and mining results are definitely in need, and the implementation of the selective generation mode is another major work. These are what we strive for in our future work.

6 CONCLUSIONS

Thousands of research papers related to pattern mining have been published so far, yet the goal of reliable real world pattern mining is still far to reach, since the simplest classic case mining has not been well pursued as revealed in this paper. The basic reason for this is the lack of well-established mining reliability theory and criteria for different mining approaches to comply with. This paper reexamines the two reliability determinants, the support $s(Z)$ and the full enumeration pattern generation mode generally used by previous approaches.

Traditionally $s(Z)$ is the only generally used criterion to evaluate a pattern, yet this measure is ill defined and causes serious probability anomaly. The full enumeration mode produces excessive and unrealizable patterns. Based on the investigation and analysis of the theoretical fallacies of previous approaches, a theoretic solution has been derived and proposed in this paper. This includes the reformulated $s'(Z)$ and the three fundamental rationality maintenance criteria that every mining approach should observe:

(1) the equilibrium condition,

(2) probability anomaly free, and

(3) the use of the feasible selective pattern generation mode.

Notice that no previous approach did or could claim the satisfaction of the above criteria. A natural conclusion is then, no previously proposed approach or algorithm, however efficient, could achieve a reliable pattern mining.

The $s'(Z)$ and the three new criteria added in will certainly improve the rationality of the mining theory and the reliability of the mining results. A direct outcome of the improvement is the great reduction of the number of result patterns (from the power set to less than linear) in the classic case mining without using any exogenous measure or constraint. The theory and solution proposed in this paper is thus featured with simplicity, rationality and effectiveness. These features are undoubtedly important for the rising big data science. These merits together imply a revolutionary change towards a more effective and more reliable pattern mining.

This paper, however, is only an initial work for peers to discuss and ultimately pursue a full set of the pattern mining theory. Further work such as the implementation of the selective pattern generation mode, the mining reliability theory, and so on, are major tasks and wait us to fulfill.

REFERENCES

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, Washington, D.C., USA, 1993.

[2] Kdnuggets (2011) Poll results: Data types analyzed/mined, 06 (2011). Retrieved June 30, 2011 from http://www.kdnuggets.com/2011/06/poll-results-data-types-analyzed-mined.html?k11n15.

[3] Kdnuggets (2012) Poll Results: Where did you apply Analytics/Data Mining. *Kdnuggets news*. Retrieved Dec 10, 2012 from http://www.kdnuggets.com/2012/12/poll-results-where-did-you-apply- analytics-data-mining.html.

[4] Heikki Mannila and Hannu Toivonen. 1996. Multiple uses of frequent sets and condensed representations. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996, pp189–194.

[5] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th VLDB Conference*, Santiago, Chile.

[6] Allan Gut. 2005. *Probability: A Graduate Course*. Springer 2005, ISBN 0387228330.

[7] Jiawei Han, Jian Pei, Yiwen Yin and Runying Mao. 2000. Mining frequent patterns without candidate generation. In *Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD'00)*, Dallas, TX, 2000, pp 1–12.

[8] Hannu Toivonen. 1996. Sampling Large Databases for Association Rules. In *Proceedings of the 22nd VLDB Conference*, Mumbai(Bombay), India, 1996, pp 134–145.

[9] Pradeep Shenoy , Gaurav Bhalotia , Jayant R. Haritsa , Mayank Bawa , S. Sudarshan , Devavrat Shah. 2000. Turbo-charging vertical mining of large databases. *ACM SIGMOD Record,Volume 29, Issue 2*, (June 2000), pp 22-23, ISSN:0163-5808.

[10] Mohammed Zaki. 2000. Scalable algorithms for association mining. *IEEE Transactions on Knowledge Data Engineering, Volume 12, Issue 3*, 2000. –390, ISSN: 1041-4347.

[11] Krishna Gade, Jianyong Wang, and George Karypis. 2004. Efficient closed pattern mining in the presence of tough block constraints. In *Proceeding of the 2004 international conference on knowledge discovery and data mining (KDD'04)*, Seattle, WA, 2004.

[12] D. T. Drewry, L. Gu, A. B. Hocking, K. D. Kang, R. C. Schutt, C. M. Taylor, J. L. Pfaltz. 2002. Current State of Data Mining, Technical Report: CS-2001-15, University of Virginia, Charlottesville, VA, USA.

[13] Nicolas Pasquier, Yves Bastide, Rafik Taouil, Lotfi Lakhal. 1999.Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th international conference on database theory (ICDT'99)*, Jerusalem, Israel, 1999, pp 398–416.

[14] Hui Xiong, Pang-Ning Tan, Vipin Kumar. 2006. Hyperclique pattern discovery. *Data Mining and Knowledge Discovery*, Volume 13, Number 2, (September 2006), pp. 219-242(24), Publisher: Springer.

[15] Unil Yun, Gangin Lee, and Kyung-Min Lee. 2016. Efficient representative pattern mining based on weight and maximality conditions. Expert Systems 33(5) (2016).

[16] Henk Tijms (2004) *Understanding Probability*. Cambridge University Press, 2004. ISBN: 0521833299.

[17] P. Billingsley (1996) *Probability and Measure*, 3rd Edition. Wiley-Interscience, 1995. ISBN-10: 0471007102.

[18] David J. Hand. 1999. Statistics and Data Mining: Intersecting Disciplines. *SIGKDD exploration, ACM SIGKDD*, volume 1, issue 1, 1999.

[19] Jiawei Han, Hong Cheng, Dong Xin, Xifeng Yan. 2007. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, Volume 15, No. 1, (2007), pp55-86. •

[20] Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, Alex T. Pang. 1998. Exploratory mining and pruning optimizations of constrained associations rules. In *Proceeding of the 1998 ACM-SIGMOD international conference on management of data SIGMOD'98)*, Seattle, WA, 1998, pp 13–24.

[21] Jian Pei, Jiawei Han and Laks V. S. Lakshmanan (2001) Mining frequent itemsets with convertible constraints. In *Proceeding of the 2001 international conference on data engineering (ICDE'01)*, Heidelberg, Germany, 2001.

[22] Brad Morantz. 2009. Constrained Data Mining. *Encyclopedia of Data Warehouse*, Volume I, by J. Wang, Second Edition. Publisher, Information Science Reference, 2009,

ISBN: 978- 1-60566-010-3.

[23] Toon Calders and Bart Goethals. 2002. Mining All Non- derivable Frequent Itemsets. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD*, 2002.

[24] Jean-Fran, cois Boulicaut, Artur Bykowski and Christophe Rigotti. 2000. Approximation of frequency queries by means of free-sets. In *Proceedings of PKDD Intentional Conference on Principles of Data Mining and Knowledge Discovery*, 2000,

[25] Guimei Liu, Jinyan Li and Limsoon Wong. 2008. A new concise representation of frequent itemsets using generators and a positive border. *Knowledge and Information Systems*, Vol. 17, Issue 1, (2008), pp 35-56,ISSN:0219-1377.

[26] Marzena Kryszkiewicz. 2001. Concise representation of Frequent patterns based on disjunction-free generators. In *Proceedings of IEEE Int. Conf. on Data Mining*, 2001.

[27] Jianyong Wang, Jiawei Han, Ying Lu, and Petre Tzvetkov. 2005. TFP: An efficient algorithm for mining top-k frequent closed itemsets. *IEEE Trans Knowl Data Eng* (2005) 17, pp652–664.

[28] Xifeng Yan, Hong Cheng, Jiawei Han and Dong Xin. 2005. Summarizing itemset patterns: a profile-based approach. In *Proceedings of the 2005 ACM SIGKDD international conference on knowledge discovery in databases (KDD'05)*, Chicago, IL.

[29] Yang Xiang, Ruoming Jin, David Fuhry and Feodor F. Dragan. 2008. Succinct summarization of transactional databases: an overlapped hyperrectangle scheme. In *Proceedings of KDD'08.*

[30] Taneli Mielikäinen. 2004. An Automata Approach to Pattern Collections. In *Knowledge Discovery in Inductive Databases, 3rd International Workshop, KDID*, 2004.

[31] Taneli Mielikäinenv. 2004. Implicit Enumeration of Patterns. In *Knowledge Discovery in Inductive Databases, 3rd International Workshop, KDID*, 2004.

[32] Chee-yong Chan and Yannis Ioannidis. 1999. An Efficient Bitmap Encoding Scheme for Selection Queries. In *Proceedings of the 1999 ACM SIGMOD international conference on management of data*, 1999.

[33] Jilles Vreeken, Matthijs van Leeuwen, Arno Siebes. 2011. Krimp: Mining itemsets that compress. *Data Mining and Knowledge Discovery*, 2011, 23(1).

[34] D.W. Cheung, Jiawei Han, V.T. Ng, C.Y. Wong. 1996. Maintenance of discovered association rules in large databases: an incremental updating technique. In *Proceedings of the 1996 international conference on data engineering (ICDE'96)*, New Orleans, LA, 1996.

[35] Sergey Brin, Rajeev Motwani, Jeffrey Ullman and Shalom Tsur. 1997. Dynamic itemset counting and implication rules for market basket analysis. In *Proceedings of the 1997 ACM-SIGMOD international conference on management of data (SIGMOD'97)*, Tucson, AZ, 1997, pp 255–264.

[36] D.W. Cheung, Jiawei Han, V.T. Ng, A.W. Fu, Yongjian Fu. 1996. A fast distributed algorithm for mining association rules. In *Proceedings of the 1996 international conference on parallel and distributed information systems*, Miami Beach, FL, 1996.

[37] Heungmo Ryang and Unil Yun. 2015. Top-K High Utility Pattern Mining with Effective Threshold Raising Strategies, Knowledge-Based Systems, 76, 109-126.

[38] Jong Soo Park , Ming-syan Chen and Philip S. Yu. 1995. An effective hash based algorithm for mining association rules. In *Proceedings of the 1995 ACM-SIGMOD international conference on management of data(SIGMOD'95)*, San Jose, CA, 1995.

[39] Ashok Savasere, Edward Omiecinski and Shamkant Navathe. 1996. An efficient algorithm for mining association rules in large databases. In *Proceeding of the 1995 international conference on very large data bases (VLDB'95)*, Zurich, Switzerland, 1995,

[40] Jin Soung Yoo and Mark Bow. 2011. Mining top-k closed co-location patterns. In *IEEE international conference on spatial data mining and geographical knowledge services (ICSDM)*, June 2011. •

[41] Guimei Liu, Hongjun Lu, Wenwu Lou and Jeffrey Xu Yu. 2003. On computing, storing and querying frequent patterns. In *Proceedings of the 2003 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03)*, Washington, DC, 2003.

[42] Gösta Grahne and Jianfei Zhu (2003) Efficiently using prefix- trees in mining frequent itemsets. In *Proceedings of the ICDM'03 international workshop on frequent itemset mining implementations (FIMI'03)*, Melbourne, FL, 2003.

[43] C. Ordonez, E. Omiecinski, L. de Braal, C.A. Santana, N. Ezquerra and J.A. Taboad (2001) Mining constrained association rules to predict heart disease. *IEEE International Conf. on Data Mining, ICDM* 2001.

[44] Charu C. Aggarwal. 2014. An Introduction to Frequent Pattern Mining. Chapter 1 of *Frequent Pattern Mining*, edited by Charu C. Aggarwal and Jiawei. Han, Springer International Publishing, 2014, Printed ISBN 978-3-319-07820-5.

[45] Stephen Stigler. 2008. Fisher and the 5% level. *Chance*, Vol. 21. No. 4, Springer New York, 2008, pp 12, ISSN: 0933-2480 (Print) 1867-2280 (Online).

[46] Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences.* 2nd ed. 1988. ISBN 0-8058-0283-5.

[47] FIMI. 2009. *Frequent Itemset Mining Dataset Repository.* Retrieved July 2009 from http://fimi.cs.helsinki.fi/data/

[48] Unil Yun, Donggyu Kim (2017) Mining of high average-utility itemsets using novel list structure and pruning strategy. *Future Generation Comp.* Syst. 68 (2017).

[49] Zhongmei Zhou, Zhaohui Wu, Yi Feng, Zhongmei Zhou, Zhaohui Wu and Yi Feng. 2006. Enhancing Reliability throughout Knowledge Discovery Process. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)*, ICDMW, 2006, pp754-758,

[50] Tongyuan Wang, Bipin C. Desai. 2009. "Issues in Pattern Mining and their Resolutions", Proceedings of Canadian Conference on Computer Science & Software Engineering, C3S2E 2009, Montreal, Quebec, Canada. ACM International Conference Proceeding Series, ACM 2009, pp17-28. ISBN 978-1-60558-401-0.

[51] Zaheer Ul-Haq and Jeffry D. Madura. 2015. Computer Applications for Drug Design and Biomolecular Systems, *Frontiers in Computational Chemistry*: Volume 2, 1st Edition, Nov. 2015. Print Book ISBN: 9781608059799, eBook ISBN: 9781608059782.

[52] John F. Lucas (1990) Introduction to Abstract Mathematics. Rowman & Littlefield. ISBN 9780912675732.

[53] Richard A. Brualdi. 2004. Introductory Combinatorics (4th ed.). Pearson Prentice Hall. ISBN 0-13-100119-1.

[54] Gregory Piatetsky-Shapiro, and Christopher J. Andmatheus. 1994. The interestingness of deviations. In *Proceedings of the AAAI-94 Workshop on Knowledge Discovery in Databases (KDD-94).* Seattle, WA. 25–36.

[55] Robert J. Hilderman and Howard J. Hamilton (2003) Measuring the interestingness of discovered knowledge: A principled approach. *Intelligent Data Analysis* 7(4).

[56] Pang-ning Tan, Vipin Kumar, and Jaideep Srivastava. 2002. Selecting the right interestingness measure for association patterns. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press.

[57] Liqiang Geng and Howard J. Hamilton (2006) Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)* 38 (3), 9, 2006

[58] Kenneth McGarry (2005) A survey of interestingness measures for knowledge discovery. *Knowl. Eng. Review* 20, 1, 39–61, 2005.

[59] Philippe Lenca, Patrick Meyer, Benoît Vaillant and Stéphane Lallich. 2004. A multicriteria decision aid for interestingness measure selection. *Tech. Rep.* LUSSI-TR-2004-01-EN, May 2004. LUSSI Department, GET/ENST, Bretagne, France.

[60] Miho Ohsaki, Shinya Kitaguchi, Kazuya Okamoto, Hideto Yokoi and Takahira Yamaguchi. 2004. Evaluation of rule interestingness measures with a clinical dataset on hepatitis. In *Proceedings of the 8th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2004).* Pisa, Italy. 362–373.

[61] Nada Lavrač, Peter Flach and Blaz Zupan. 1999. Rule evaluation measures: A unifying view. In *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP '99).* Bled, Slovenia. Springer-Verlag, 174–185.

[62] Martin Kirchgessner, Vincent Leroy, Sihem Amer-Yahia and Shashwat Mishra. 2016. Testing Interestingness Measures in Practice: A Large-Scale Analysis of Buying Patterns. *Computing Research Repository*, 2016, Volume abs/1603.04792.

[63] Fabrice Guillet and Howard J. Hamilton (Eds.). 2007. Quality Measures in Data Mining. *Studies in Computational Intelligence*, 2007, Volume 43. ISBN 3-540-44911-6.

[64] M. Padmavalli, K. Sreenivasa Rao (2013) An Efficient Interesting Weighted Association Rule Mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 10, October 2013 ISSN: 2277 128X.

[65] Haoran Zhang, Jianwu Zhang, Xuyang Wei, Xueyan Zhang, Tengfei Zou and Guocai Yang. 2017. A New Frequent Pattern Mining Algorithm with Weighted Multiple Minimum Supports. *Intelligent Automation & Soft Computing*, 23:4, 605-612, DOI: 10.1080/10798587.2017.1316082.

[66] D. Sujatha and Naveen C. H. (2011) Quantitative Association Rule Mining on Weighted Transactional Data, *International Journal of Information and Education Technology*, Vol. 1, No. 3, August 2011.

[67] Bay Vo, Frans Coenen and Bac Le. 2013. A new method for mining Frequent Weighted Itemsets based on WIT-trees, *Expert Systems with Applications*, Volume 40, Issue 4, March 2013, Pages 1256-1264, https://doi.org/10.1016/j.eswa.2012.08.065.

[68] Anshu Zhang, Wenzhong Shi and Geoffrey I. Webb. 2016, Mining significant association rules from uncertain data. (12 January 2016) *Data Mining and Knowledge Discovery*, 10.1007/s10618-015-0446-6.

[69] Jerry Chun-Wei Lin, Wensheng Gan, Philippe Fournier-Viger, Tzung-Pei Hong and Han-Chieh Chao. 2017. Mining Weighted Frequent Itemsets without Candidate Generation in Uncertain Databases. *International Journal of Information Technology & Decision Making*, 2017, Volume 16, Number 06, Page 1549, DOI: 10.1142/S0219622017500341.

[70] Raymond A. Serway, Robert J. Beichner and John W. Jewett,Jr. 2000. Physics for Scientists and Engineers, Saunders College Publishing. ISBN 0-03-022654-6

[71] Bakshi Rohit Prasad and Sonali Agarwal. 2016. Stream Data Mining: Platforms, Algorithms, Performance Evaluators and Research Trends. *International journal of database theory and application*, Vol. 9, No. 9 (2016), pp 201-218, DOI: 10.14257/ijdta.2016.9.9.19

[72] Shikha Mehta Janardan (2017) Concept drift in Streaming Data Classification: Algorithms, Platforms and Issues. *Information Technology and Quantitative Management (ITQM 2017)*, Procedia Computer Science, Volume 122, 2017, Pages 804–811, Elsevier.

A New Malware Detection System Using a High Performance-ELM method

Shahab Shamshirband[1]

[1]Department of Computer and Information Science, Norwegian University of Science and Technology Norway

shahab.shamshirband@ntnu.no

Anthony T. Chronopoulos[2,3]

[2]Department of Computer Science, University of Texas at San Antonio, USA

[3] (Visiting Faculty) Dept of Computer Engineering & Informatics, University of Patras, Greece

Anthony.Chronopoulos@utsa.edu

ABSTRACT

A vital element of a cyberspace infrastructure is cybersecurity. Many protocols proposed for security issues, which leads to anomalies that affect the related infrastructure of cyberspace. Machine learning (ML) methods used to mitigate anomalies behavior in mobile devices. This paper aims to apply a High-Performance Extreme Learning Machine (HP-ELM) to detect possible anomalies in two malware datasets. Two widely used datasets (the CTU-13 and Malware) are used to test the effectiveness of HP-ELM. Extensive comparisons are carried out in order to validate the effectiveness of the HP-ELM learning method. The experiment results demonstrate that the HP-ELM was the highest accuracy of performance of 0.9592 for the top 3 features with one activation function.

CCS CONCEPTS: Security and privacy →Intrusion/anomaly detection and malware mitigation

Keywords:

Learning, Malware Detection, High Performance Extreme Learning Machine (HP-ELM), Mobile Apps, Internet of Thing (IoT).

ACM Reference Format:

Shahab Shamshirband, Anthony T. Chronopoulos. 2019. Malware Detection System using High Performance-ELM method. In IDEAS 2019: 23rd International Database Engineering & Applications Symposium, June 10–12, 2019, Athens, Greece. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331076.3331119

1 Introduction

The popularity of smartphones is growing recently. According to the Gartner, the total number of smartphone's user will reach six billion by 2020 (according to Ericsson [1]), also more than 40 million attacks by malicious mobile malware (according to Kaspersky Labs [2]), and a data breach will exceed $150 million by 2020 (according to Juniper research [3]). Smartphone devices are affected by malicious malware which installs their modules in the system directory and attempts to subvert the typical system's behavior [4].

To cope with these problems, researchers and security analysts conducted a study for practical or scientific purposes to establish reliable applications for mobile devices. In [5], Russon discovered various types of hidden malware in more than 104 Google play applications which downloaded over 3.2 million times. It causes numerous attacks to user's mobile devices to affect the CPU loads for the system [5].

Several security protocols applied for malware detection. Such systems can be either anomaly-based or behavior-based. The former relies on a predefined pattern. This type of Intrusion Detection System (IDS) is an efficient approach for identifying sweeps and probes of network hardware and hints early warnings of potential intrusions before firing attacks such as Telnet. Such type systems depend on receiving regular signature updates (i.e., the extent of the signature database). The dramatic influence of DDoS attacks by Mirai bot net and its variants highlights the risks for IoT devices [6]. An end to end security scheme called Datagram Transport Layer Security (DTLS) protocol utilizes an encryption technique [7]. Although DTLS mitigate the specific type of attacks, it fails to identify all type of big data based anomalous behaviors. The significant drawback is that they are attack-specific [8]. Therefore, it is essential to develop methodologies and procedures to measure the uncertainty of IoT devices and its potential capacity to make a smart decision in order to increase the efficiency of security and privacy issue in IoT environments. It is also essential to provide smart recommendation in terms of fraud or vulnerability detection using leaning algorithms.

Soft computing techniques such as neuro fuzzy have been proposed to mitigate the security issue in IoT based malware detection [9]. Neuro-Fuzzy classifiers trained from static data and data that applications generate during their execution. Neuro-Fuzzy has a potential in malware detection based on collected statistics and derived fuzzy rules ([10], [11], [12]). Moreover, several learning algorithms have been proposed to identify the malware codes and their behaviors and identify the specific type of threats ([13],[14]). The main drawback of soft computing technique such as neuro fuzzy is that the fuzzy rules are randomly generated to tune the weights of the neural network and the number of hidden layers can increase depending on the type of application. However, there is no way to speed up the procedure of tuning. Thus, ELM can adjust the activation functions in the Single Hidden Layer Feed-forward Neural Networks (SLFNs) ([15] , [16] , [17]) and [18].

In this paper, we utilized a fast convergent reinforcement solution named High Performance Extreme Learning Machine (HP-ELM) to help the learning phase of the method to call the parameters of the hidden neurons that created randomly, which is independent of the training data [19]. Furthermore, such a method is used to test various scales of data sets, different structure selection options, and regularization methods. In our study, HP-ELM is applied to classify the malware in two datasets.

In our study, we deal with the following research questions: 1. “What are the current data analytic techniques that are being used to extract meaningful IoT based malware devices values?”, 2. “How to maintain malware influence on IoT?” and 3. “What is the effect of the proposed security and privacy preserving framework in terms of scalability in the big data platform?”. Our work contributes to forensic malware behaviour. Previous results and datasets of mobile malware applications used in this study ([20] ,[21]).

The contribution of the paper is to propose an IDS method to recognize IoT malware as follows. We applied a feature selection method in two malware datasets. We developed a malware detection system for IoT environment based on HP-ELM classifier. We tested the efficiency of the proposed system using two benchmark datasets: CTU-13 dataset [20] and DyHAP malware dataset [21] and we compared the performance of HP-ELM with and without feature selection.

The manuscript organized as follows. The previous works discussed in Section 2. Section 3 presents the HP-ELM detection method and Section 4 discusses data preparation. Section 5 presents scenarios and the setting of the HP-ELM parameters. Section 6 presents the simulation setup and evaluation metrics — experiments presented in Section 7. Section 8 concludes the paper and presents future directions.

2 Previous work

This section presents technical papers which use the security framework for malware based IoT [22]. In [22], an application program is tested in a scrutinizing manner without the implementation of the actual application (i.e., a reverse engineer process) while in [23], a program investigates the behavior of the running processes by executing the application. On the one hand, a static type malware program requires low memory resources, minimal CPU processes and the analysis process is fast, on the other hand, a dynamic one could be used to detect unknown changes and malware existence [23]. A machine learning based malware detection proposed in [24] which used a private and a public dataset. Finally, they validated their solutions in various cases. In our paper, we also evaluated the proposed methods in a private (Andoird Malware) and a public (CTU-13) dataset using a different type of HP-ELM parameter setting experiments.

Authors in [23] use a crowdsourcing system in order to attain the flows of the application’s behavior. Many studies have been carried out on mobile phone malware based on a single operating system or a comparative study between two operating systems. Authors in [22] present a multilevel and behavior-based Android malware detection using 125 existing malware families and report 96% detection of malware. However, this approach applies system calls which contains less semantic information and is not able to detect malicious behavior accurately. Recently, authors in [25] captured system calls and “binder transactions” which runs the runtime behavior signature. This method presents a new malware variant detection client-server system which jointly covers the logic structures and the runtime behaviors of mobile applications for Android devices. However, this approach depends on the network status, graph mining; it impacts network performance, which directly influences the computation complexity. Therefore, it is not suitable for real-time detection.

In [26] authors present a lightweight detection system to identify malicious behaviors from mobile devices. Statistical Markov chain models applied to build the application behavior in the form of a feature vector and the random forest is adopted to classify the application behavior. The results indicate an accuracy of 96%.

Mirai attempts to categorize remarkable DDoS attacks affecting high profile targets [6]. It is a wake-up call for the control in IoT devices and analyzes the risk of increasingly DDoS attacks. It diminishes the administrative credentials of IoT devices using brute force, relying on a small dictionary of a possible username and password pairs [27]. To cope with such issues, we need resource efficiency hybrid IDSs as novel security solutions that can efficiently protect devices against DDoS, considering the insufficient resources available in the IoT environment.

3 HP-ELM detection method

The main drawback of soft computing technique is that the fuzzy rules are randomly generated to tune the weights of the neural network and the number of hidden layers can increase depending on the type of application. However, there is no way to speed up the procedure of tuning. Thus, ELM able adjust the activation functions in the single hidden layer feed-forward neural networks (SLFNs) ([15] , [16] , [17]) and [18].

This section presents the HP-ELM mobile network architecture applied on two malware datasets Android Malware [21] and CTU-13 dataset [20]. As mentioned in Figure 1, ELM is fast training method for SLFN networks[28].

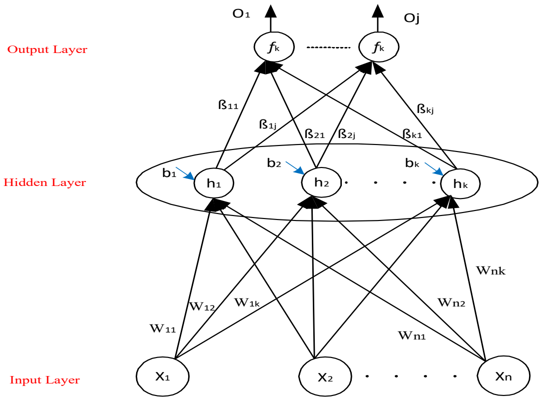


Figure 1. Computing the output of an SLFN (ELM) model

Figure 1 shows the computing of output of an ELM model. It includes three layers of neurons. There is no computation in layer one (input). The input layer weights ω and biases b are set randomly and never adjusted (random distribution of the weight). The output layer is linear and there is no transformation function and bias for output layer. Therefore, the computing time is very fast. The word "Single" in "SLFN" is because there is only one layer of non-linear neurons (hidden layer). The main advantage of ELM is to produce weakly connected hidden layer features, because the input layer weights randomly generated, and it improves the generalization properties of the solution of a linear output layer [19].

The ELM method described as follows. We consider a set of N distinct training samples (x_i, t_i) , $i \in [1, N]$ with $x_i \in R^c$ and $t_i \in R^c$. A SLFN with L hidden neurons has the following output equations:

$$\sum_{j=1}^{L} \beta_j \emptyset(w_j x_i + b_j), \quad i \in [1, N], \tag{1}$$

where $\emptyset$ is the activation function (a sigmoid function is a common choice, but other activation functions are possible including linear [17], [18] and [28]), $w_j = \left[w_{j1,} w_{j2,} w_{jn,}\right]^T$ is the weight vector that connects the n input nodes to the jth hidden node, b_i are the biases values of the jth node.

A hidden node and $\beta_j = \left[\beta_{j1,} \beta_{j2,} \beta_{jm,}\right]^T$ is the set of values of the output weights that connects the jth hidden node with m output nodes. The relation between inputs x_i of the network, target outputs t_i and estimated outputs y_i is:

$$y_i = \sum_{i=1}^{L} \beta_j \emptyset(w_j x_i + b_j) = t_i + \epsilon_i, \quad i \in [1, N], \tag{2}$$

where $\emptyset$ is the activation function and w_j is the weight vector that connects the n input nodes to the jth hidden node, b_i the biases values of the jth node. The noise (ϵ) includes both random noise and dependency on variables not presented in the inputs X.

For N samples, the N equations represented as Hβ=T, where

$$H = (w_1, w_2, \dots, w_k, b_1, b_2, \dots, b_k, x_1, x_2, \dots, x_N) \tag{3}$$

$$= \begin{pmatrix} \emptyset(w_1.x_1 + b_1) & \cdots & \emptyset(w_k.x_1 + b_k) \\ \vdots & \ddots & \vdots \\ \emptyset(w_1.x_N + b_1) & \cdots & \emptyset(w_k.x_N + b_k) \end{pmatrix}_{N*k},$$

$$\beta = [\beta_1^T \dots \beta_k^T]_{k*m}^T \tag{4}$$

$$T = [y_1^T \dots y_k^T]_{N*m}^T \tag{5}$$

The output weight matrix β found by solving the least square problem:

$$\dot{\beta} = min_{\beta} \|\mathrm{H\beta - T}\| = H^{\dagger} T, \tag{6}$$

$$\dot{\beta} = (H^{\mathrm{T}} H)^{-1} H^T T \tag{7}$$

where $H^{\dagger}$ is the MP pseudo inverse of the hidden layer output matrix H.

This paper used HP-ELM toolbox which supports multi-class, weighted multi-class and multi-label as a classifier [19]. Section 4 describes how HP-ELM adapted to the two malware scenarios.

4 Dataset preparation

In this study, the HP-ELM methods evaluated on two scenarios. The first scenario uses the Mobile Malware dataset and the second scenario uses the CTU-13 dataset. We will describe the method of data collection for these two scenarios next.

4.1 Mobile Malware dataset (Scenario 1):

This scenario consists of two types of applications such as benign and malware. Twenty normal apk file downloaded from Google Play. These benign files installed on an Android-based operating system Jelly Bean version 4.3 which runs on mobile devices.

After installation, the network traffic of running apps captured in a real time network environment in order to authenticate the behavior of apps. In the case of malware, the experiment utilizes Malgenome [29] as the malware dataset. It contains 1260 samples which consist of 49 malware families used in the previous study [21]. The identification includes several malware types, such as a botnet, root exploit, and Trojan. Authors selected seven connection-based features on the Information Gain (IG) [30] algorithm to analyze it because of its practical measuring features [31]. IG shows their strength in accuracy enhancement, the capability of generalization and short execution time. It determines how the training sets separated according to the target classification [32]. In Scenario 1, the higher gain ratio indicates the feature's relevance in a classification model for a machine learning classifier. Therefore, the features are maximum_frame, frame_STD, count_ACK, Minimum_Dest_Port, Average_Frame, and Average_source_port [21]. Figure 2 shows the methodology of collection of mobile malware dataset in Scenario 1.

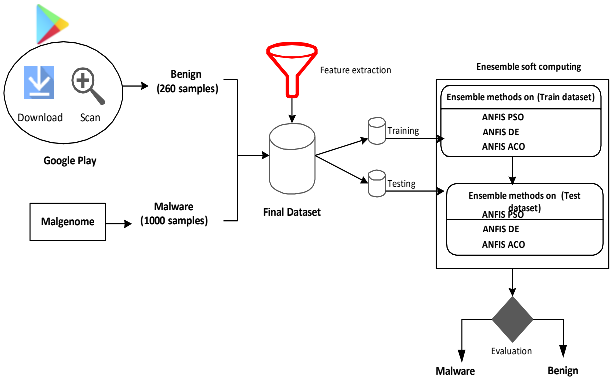


Figure 2: The methodology of collection of Mobile Malware data

4.2 CTU-13 dataset (Scenario 2):

This dataset consists of thirteen captures (called Scenarios) of different real botnet samples [20]. The characteristics of the scenarios and the features (pcap files) captured in terms of tcpdump. The pcap files are converted to netflow file standard using the argus software suite in two steps. The first step converts the pcap files to a bidirectional argus binary storage. The second step converts the argus bin to Netflow. The outcome of these steps is the final netflow file [33]. The next step is to assign the label to the netflow data. The background label assigns the normal label to the traffic which matches a specific filter. Then the botnet label is assigned to the traffic that comes from or to any of the known infected IP address [20]. Figure 3 indicates the methodology of collection of CTU-13 and how the HP-ELM classifies it.

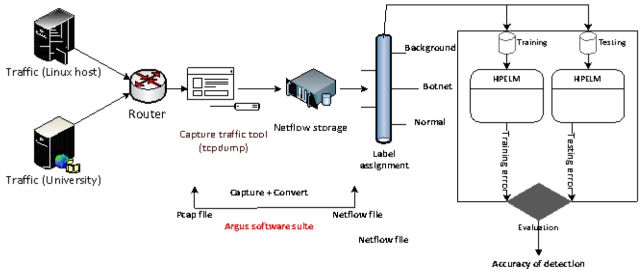


Figure 3: The methodology of collection of CTU-13

In this research work, the labeled dataset splits into training (70%) and testing data (30%) [34] to evaluate with the HP-ELM algorithm. We also evaluated our methods with different training and testing data such as 80% training and 20% testing, and we found that the best result reached is by 70% training and 30% testing. Thus, we focus only on the case with training (70%) and testing data (30%). In the next section, we propose the procedure of applying the method to the datasets.

5 Proposed method

The proposed method includes a few modules. In detail, the first module (Reading) is used to read the malware datasets continuously and transfer them to feature selection modules. Then, the second module (Filtering) is responsible for normalizing the data and applying the feature selection algorithm to select the essential features for the proposed algorithm. After that, the third module (Splitting) is responsible for splitting the data into 70% training and 30% testing. Then, the fourth module (HP-ELM) is adapted to tune its parameters to predict the targets. Finally, the fifth module (Evaluation) is responsible for predicting the actual values in the training and testing phase.

5.1 Feature selection

Before applying the HP-ELM method, we utilized the F-Score [35] and Fisher Score [36] in order to select the most valuable features. Then, the HP-ELM builds to new data. The extracted features from CTU-13 dataset are: {DstAddr, sport, Proto, SrcAddr, Dport, Dur, State, sTos, dTos, TotBytes, SrcBytes and TotPkts}c [37] . More details about the features of CTU-13 mentioned in [20].

5.2 HP-ELM parameter setting (activation functions, number of neurons, etc)

We applied six types of activation functions in the layer of HP-ELM such as linear, rbf-linf, rbf-l1, rbf-l1, tanh, and sigmoid. A total of 2000 of neurons applied in the layer of HP-ELM. We investigated HP-ELM with feature selection (Top 3, and top 5 features) and without feature selection (all features). We also applied different strategies for activation functions in the layer of HP-ELM (one, two, three and four activation functions) in both datasets.

6 Performance Evaluation

The following subsections describe in detail the pursued datasets, evaluation metrics, scenarios and their settings, and the results for the applied scenarios.

6.1 Simulation Setup

The methods tested on a system with Intel Core i7CPUand 8-GB RAM. Two benchmark malware datasets CTU-13 dataset (Scenario 5) [20] and DyHAP malware dataset [21] have been used to evaluate the performance of HP-ELM.

6.2 Metrics

This research used the accuracy ratio of Equation 8 to evaluate the performance of the method. In general, the accuracy is the number of applications which the classifier correctly detects, divided by the total number of malicious and legitimate applications. The accuracy is between 0 and 1, i.e. $0 \le Accuracy \le 1$. We also have $Accuracy\ Ratio = \frac{TP+TN}{TP+FN+TN+FP}$ (8) where the used parameters denote the following numbers. TN is the accurately classified benign instances; TP is the malicious applications that are appropriately identified, FP is the wrongly classified benign instances as malware applications; and, FN is the malware instances wrongly classified as a legitimate application.

7 Results and discussion

We evaluated the results using various layers with/without the feature selection.

7.1 Evaluate HP-ELM without applying feature selection in android malware

This subsection compared the performance of the HP-ELM in the presence of various active functions for two datasets without using feature selections and integrated all features. A total number of 13 features are available in CTU-13 dataset, and the number of total input features for Android Malware dataset is six. Figure 4 (a, b) indicates the accuracy of HP-ELM ELM in training (70%) and testing (30%) with and without feature selection with one, two and three activation function for android malware dataset. The x-axis represents the activation function's name. For example, the order of one activation function with all features are linear, Rbf_l1, Tanh, Sigmoid, Rbf_l2 and Rbf_linf which are mapped to the tag point (0.0, 0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0). It is obvious that HP-ELM reach a high accuracy of 0.9696 in training and 0.9672 in testing with 2000 neurons of Rbf-linf activation function without applying the feature selection. The result indicates that the accuracy of HP-ELM increased with two activation functions (Rbf_linf(1000), Sigmoid(1000)). Finally, the accuracy of HP-ELM with three activation functions Sigmoid (400), Tanh (1000), Rbf_linf(600), and a total number of 2000 neurons reach 0.9679 in training and 0.9661 in testing.

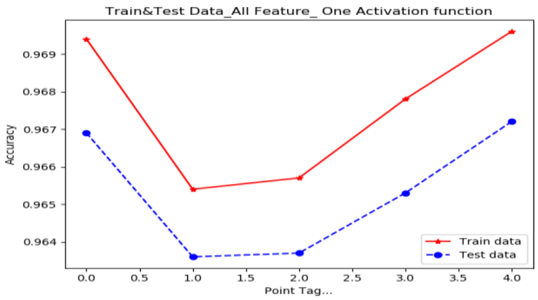


Figure 4 (a). Accuracy of HP-ELM without feature selection with one activation function for android malware dataset

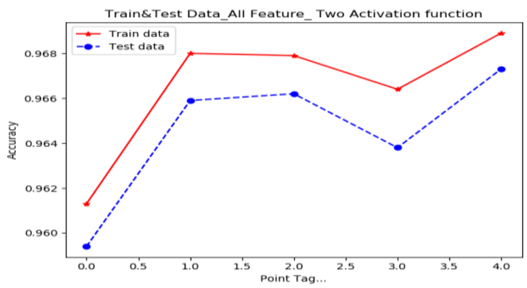


Figure 4 (b). Accuracy of HP-ELM without feature selection with Two activation function for android malware dataset

7.2 Evaluate HP-ELM by applying feature selection (Top 3 features) on android malware

This subsection compares the performance of the HP-ELM in the presence of various active functions for the android malware datasets with three higher priority features using F-Score selection policy. The goal of this scenario is to analyze the accuracy rate of the HP-ELM method in the presence of different activation functions using various selected features. In Figure 5 a and b, we have numerically tested this, by applying the top 3 features, the accuracy of HP-ELM in terms of one activation function Rbf_l1(2000) reach a high value of 0.9056 in training and 0.9017 in testing. On the other hand, a combination of two activation functions such as Tanh (1000), Rbf_l1(1000) Hp-elm gives a better accuracy ratio of 0.9018 in testing. Finally, the high accuracy reached with three activation function Sigmoid (400), Tanh (1000), Rbf_linf(600) in testing 0.8998. It can be considered the best strategy rule of a two-activation function. Figure 5a, and 5b indicate the accuracy of HP-ELM in android malware scenario.

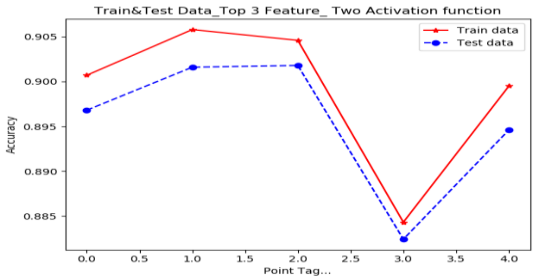


Figure 5 (a). Accuracy of HP-ELM with top 3 features and two activation function for android malware dataset

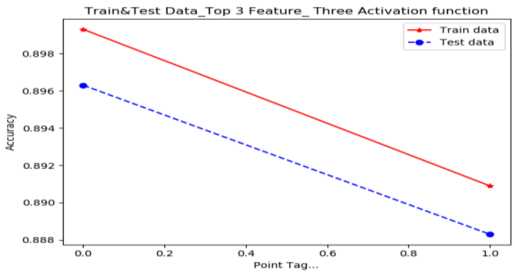


Figure 5 (b). Accuracy of HP-ELM with top 3 feature with three activation function for android malware dataset

7.3 Evaluate HP-ELM with applying feature selection (Top 5 features) on android malware

This subsection compares the performance of the HP-ELM in the presence of various active functions for the same layer for android malware datasets with five higher priority features using F-Score selection policy. The goal of this scenario is to analyze the accuracy rate of the HP-ELM method in the presence of different activation functions using various selected features. In Figure 6 a and b, we have tested the model by applying top 5 features. The accuracy of HP-ELM in terms of one activation function Rbf_linf (2000) reach to the high value of 0.9675 in training and 0.9656 in testing. However, the testing result of Rbf_l1 (2000) is higher than Rbf_linf (2000), which is 0.9702. The result of a combination of two activation functions Rbf_linf (1000), Sigmoid (1000), Hp-elm gives a better accuracy ratio of 0.9673 in training and 0.9654 in testing, but the combination of two activation functions of Rbf_l1 (1000), Sigmoid (1000) reaches a high accuracy in testing phase which is equal to 0.9689. The combination of three activation functions such as Sigmoid (400), Tanh (1000), Rbf_linf(600), gives the accuracy of 0.9668 in training and 0.9661 in testing which is not very high in comparison with the cases of two and one activation functions.

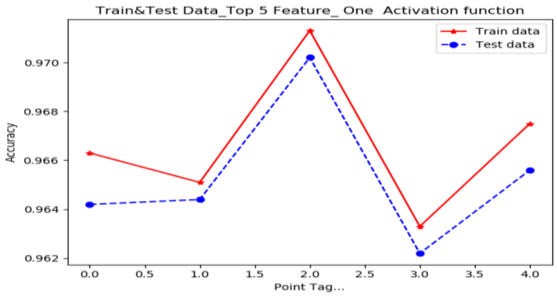


Figure 6 (a). Accuracy of HP-ELM with top 5 features and one activation function for android malware dataset

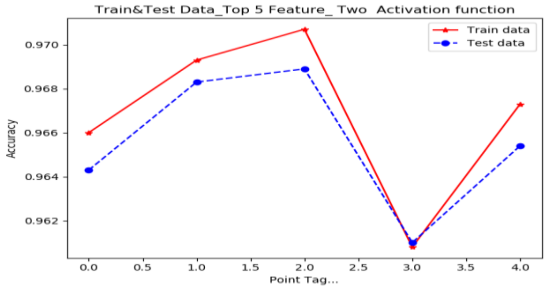


Figure 6 (b). Accuracy of HP-ELM with top 5 features and two activation function for android malware dataset

In Figure 6, we confirm that when the number of activation function increases, the accuracy will also increase, and its value will remain constant after 1000 neurons in datasets. By increasing the number of activation (two) functions, the highest accuracy reached in the testing phase.

7.4 Evaluate HP-ELM without applying feature selection on CTU-13

HP-ELM applied to CTU-13 in training and test cases. For example, with one activation function (Rbf_linf) with 2000 neurons reaches a high accuracy of classification of 0.9625 in testing. On the other hand, the higher accuracy with two activation functions rbf_l1(1000), rbf_linf(1000) is 0.9622 in the testing phase. By increasing the number of activation functions to three, the result does not reach as high accuracy in comparison with one and two activation functions. The highest accuracy with three activation functions is 0.9616 which is lower than 0.9625 obtained with one activation function and 0.9622 obtained with two activation function. According to Figures 7 (a) ,7 (b) and 7(c) for the CTU-13 dataset, it is confirmed that more activation functions provide more complexities. It observed that the rate does not increase. Moreover, when we reach to the best number of neurons, increasing activation functions does not affect the accuracy in comparison with the case of one activation function in the layer of HP-ELM.

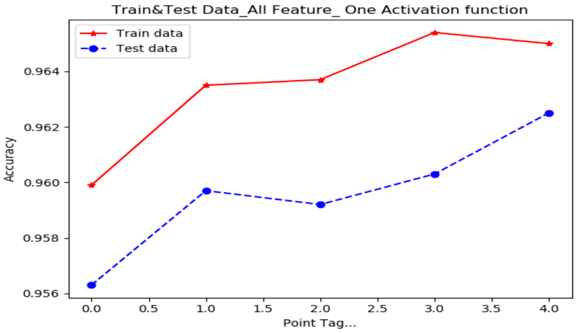


Figure 7 (a). Accuracy of HP-ELM without feature selection and one activation function for CUT-13

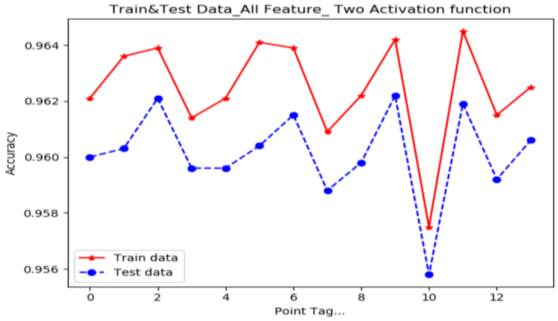


Figure 7 (b). Accuracy of HP-ELM without feature selection and Two activation function for CUT

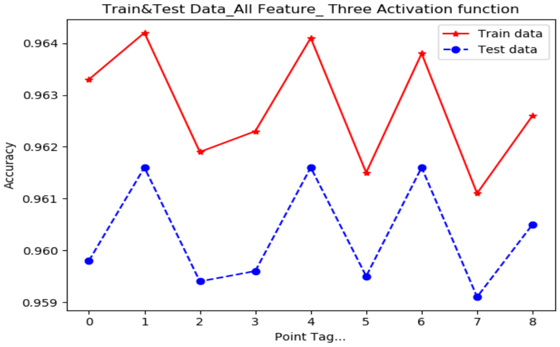
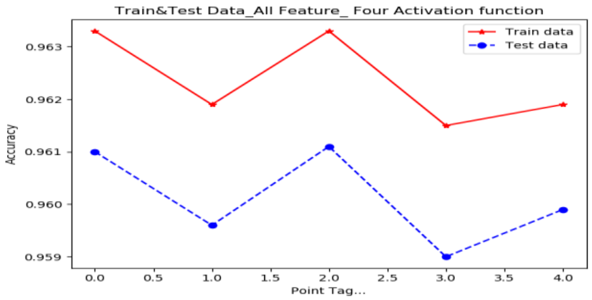


Figure 7 (c). Accuracy of HP-ELM without feature selection and Three activation function for CUT-13

Figure 7 (d). Accuracy of HP-ELM w



ithout feature selection and Four activation function for CUT-13

7.5 Evaluate HP-ELM with applying feature selection on CTU-13 (Top 3)

In this section, the feature selection strategy applied to CTU-13. Therefore, three high priority features are selected based on the F-score algorithm, then the selected feature allows hp-elm to evaluate the performance of the method. As in previous sections, various activation functions with a total number of 2000 of neurons applied to CTU-13. On the other hand, the highest accuracy with one activation function (Rbf-linf) reaches 0.9592 in the testing phase. Figure 8 (a,b,c) shows the accuracy of hp-elm with the top 3 features and different activation functions.

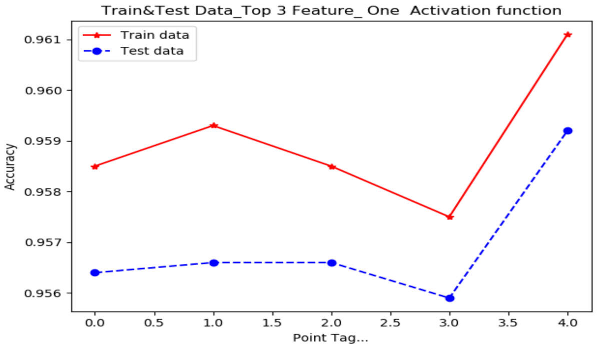


Figure 8 (a). Accuracy of HP-ELM with feature selection and one activation function for CUT-13

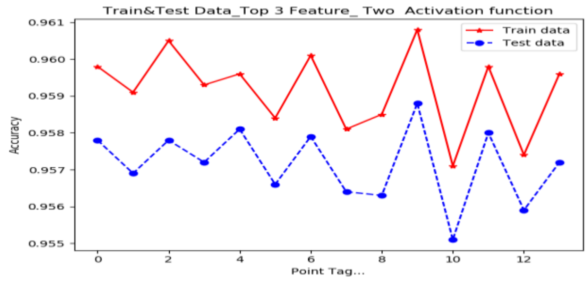


Figure 8 (b). Accuracy of HP-ELM with feature selection and two activation function for CUT-13

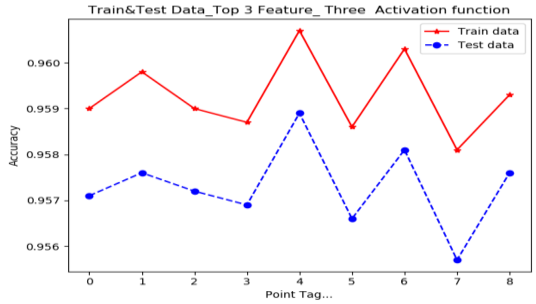


Figure 8 (c). Accuracy of HP-ELM with feature selection and three activation function for CUT-13

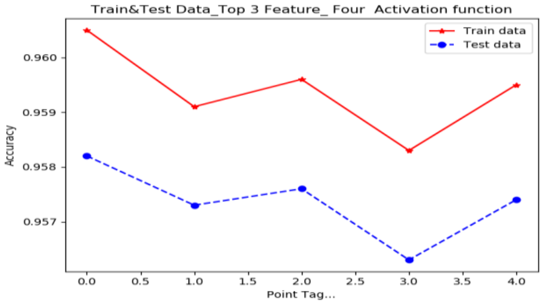


Figure 8 (d). Accuracy of HP-ELM with feature selection and four activation function for CUT-13

7.6 Evaluate HP-ELM with applying feature selection on CTU-13 (Top 5)

In this scenario, five high priority features are selected based on the F-score algorithm. By applying one activation function with a total number of 2000 of neurons, high accuracy of 0.9589 in Rbf_linf (2000) is reached in the testing phase as shown in Figure 9 a-d.

The second-highest accuracy with two activation function of rbf_linf(1000), linear(1000) is 0.9590 in the testing phase. On the other hand, the third highest accuracy with three activation function rbf_11(666), rbf_112(667), rbf_linf (667) is 0.9588 in testing phase. By increasing the number of activation functions to four, the accuracy of 0.9585 is the same in the testing phase for the activation function of 1) Tanh(500), sigm(500), rbf_11(500), rbf_linf (500) 2) sigm(500), rbf_l1(500), rbf_l2(500), linear (500) and 3) rbf_l1(500), rbf_l2(500), rbf_linf (500) linear (500), but the training phase of Tanh(500), sigm(500), rbf_11(500), rbf_linf (500) is better than the rest of the activation functions.

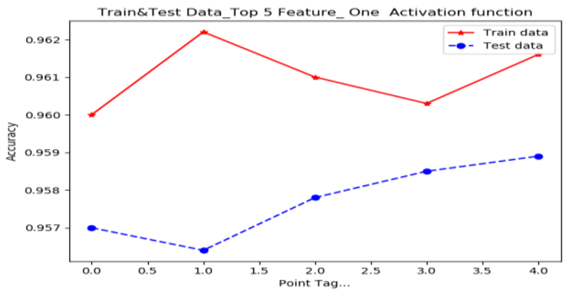


Figure 9 (a). Accuracy of HP-ELM with feature selection and one activation function for CUT-13

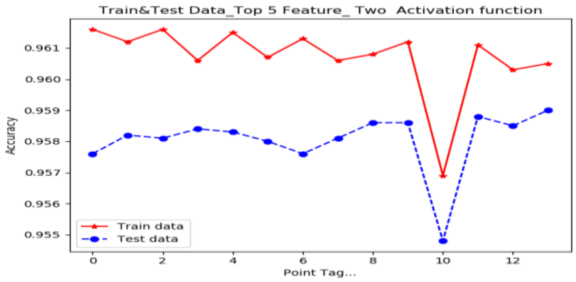


Figure 9 (b). Accuracy of HP-ELM with feature selection and two activation function for CUT-13

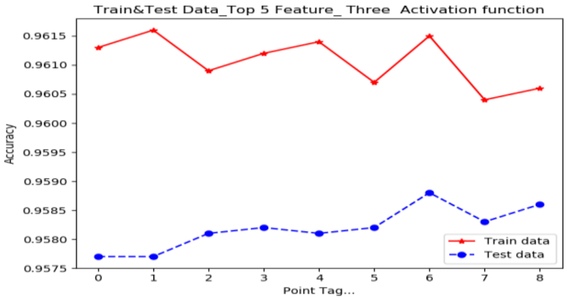


Figure 9 (c). Accuracy of HP-ELM with feature selection and three activation function for CUT-13

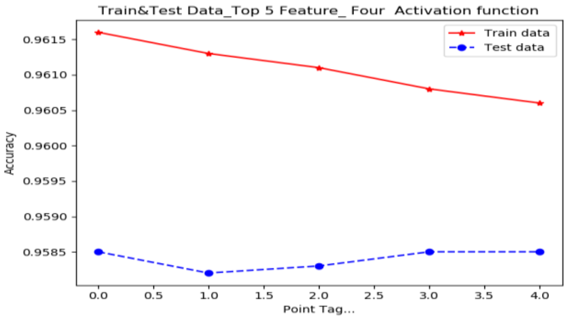


Figure 9 (d). Accuracy of HP-ELM with feature selection and four activation function for CUT-13

7.7 Comparison of results

For the evaluation of a proposed model, the key point is to show that the HP-ELM model is producing the smallest error in testing datasets or the highest accuracy of classification. As Table 1 shows, the accuracy values for malware dataset without feature selection are rather high with two activation function Rbf_linf (1000), and Sigmoid (1000). From Table 2 with feature selection, it is apparent that the precision is rather high in terms of two activation function of Rbf_l1 (1000) and Sigmoid (1000) compared with other metrics. The outcome of Table 3 and 4 indicate that the higher accuracy of HP-ELM in CTU-13 without feature selection is 0.9625 in the testing phase which is related to one activation function Rbf_linf(2000). On the other hand, the best accuracy of feature selection based on HP-ELM for top 3 is 0.9592 which is Rbf_linf(2000) and the higher accuracy for top 5 feature in testing is 0.9590 which is rbf_linf(1000), linear(1000).

Table 1: Accuracy of HP-ELM without feature selection in malware dataset

	Without feature selection			
Dataset	#Activation function		All features	
			Train	Test
Malware	1 activation function	Rbf_linf(2000)	0.9696	0.9672
	2 activation function	Rbf_linf(1000), Sigmoid(1000)	0.9689	0.9673
	3 activation function	Sigmoid (400), Tanh (1000), Rbf_linf(600),	0.9679	0.9661

Table 2: Accuracy of HP-ELM with feature selection in malware dataset

	With feature selection						
Data	Act- func		Top 3		Act-func	Top 5	
			Train	Test		Train	Test
Malware	1 act func	Rbf_l1 (2000)	0.9056	0.9017	Rbf_linf (2000)	0.9675	0.9656
	2 act func	Rbf_l1 (1000), Sigmoid (1000)	0.9046	0.9018	Rbf_l1 (1000), Sigmoid (1000)	0.9707	0.9689
	3 act fun	Sigmoid (400), Tanh (1000), Rbf_linf(600)	0.9037	0.8998	Sigmoid (400), Tanh (1000), Rbf_linf(600)	0.9668	0.9661

The Table 3 and 4 reports summarize information on the total number of evaluations in malware dataset which gained accuracy for HP-ELM in two experiments, with and without feature selection, which specifically focus on the number of activation functions. In the feature selection experiment, the total numbers of activation functions remained the same, but there was a significant difference in the accuracy of classification by increasing the number of feature selection. For instance, the accuracy of testing (0.9689) in the top 5 features is higher than the accuracy of classification (0.9018) with the top 3 feature selection in the same activation function and the number of neurons. As shown in Table 2, in malware experiments with three activation functions, out of a total of 2000 neuron, the percentage of accuracy in top 5 feature was higher. In general, it can be seen that by increasing the number of activation functions without feature selection strategy, the percentage of accuracy is gradually dropping. On the other hand, by selecting more features and activation functions, the accuracy of classification increases moderately.

Table 3: Accuracy of HP-ELM without feature selection in CTU-13 dataset

	#Activation function	Without feature selection		
Datas et			Train	Test
CTU-13	1 activation function	Rbf_linf(2000)	0.965	0.9625
	2 activation function	rbf_l1(1000), rbf_linf(1000)	0.9642	0.9622
	3 activation function	Tanh (666), sigm(667), rbf_linf(667)	0.9642	0.9616
	4 activation function	sigm(500), rbf_11(500), rbf_12(500), rbf_linf (500)	0.9633	0.9611

Table 4: Accuracy of HP-ELM with feature selection in CTU-13 dataset

		With feature selection				
#Activation function		Top 3		#Act function	Top 5	
CTU-13		Train	Test		Train	Test
1 act function	rbf_lin f(2000)	0.9611	0.9592	rbf_linf(2000)	0.9616	0.9589

2 act function	rbf_l1 (1000), rbf_linf(1000)	0.9608	0.9588	rbf_linf(1000)
				,
				linear(1000)
					0.9605	0.9590

3 act function	Sigm(666), rbf_11(667), rbf_linf(667)	0.9607	0.9589	rbf_11(666)
				rbf_l12(667)
				rbf_linf(667)
					0.9615	0.9588

4 act function	Tanh (500), sigm(500), rbf_11(500), rbf_linf (500)	0.9605	0.9582	Tanh (500)
				sigm(500)
				rbf_11(500)
				rbf_linf (500)
					0.9616	0.9585

Table 3 and 4 gives information on the accuracy of classification of HP-ELM for two experiments, with and without feature selection of CTU-13 dataset. In the feature selection experiment, HP-ELM was the highest accuracy leader with a low number of feature selection, and it is about 0.9592 for the top 3 features with one activation function.

8 Conclusion and future discussion

In this paper, we exploit HP-ELM to improve prediction stability for the training of (SLFNs). The presented model optimizes the input weights and hidden layers and provides more consistent performance in comparison to the other training models. Actual performance of the HP-ELM classifier tested under two real-world datasets, with various activation functions, neurons, layers, and different feature selections. The simulation results show that HP-ELM is a fast training method that reduces the root mean square error near to zero, it approaches accuracy ratio to one, and it achieves the feature optimization combination, and it also provides an excellent generalization performance on an SLFN and establishes a network intrusion detection model with the best overall performance. This finding, however, shows that despite promising results obtained by using the proposed model for this case study, it improved and further studies, which take more variables into account, will need to be undertaken.

Acknowledgments

- This work has been supported by the Norwegian Open AI Lab. The authors would like to thank Keith Downing for valuable technical comments.

References

[1] P. Cerwall, P. Jonsson, R. Möller, S. Bävertoft, S. Carson, I. Godor, P. Kersch, A. Kälvemark, G. Lemne, and P. Lindberg, “Ericsson mobility report,” On the Pulse of the Networked Society. Hg. v. Ericsson, 2015.

[2] "Android Mobile Security Threats."

[3] S. Smith, "Cybercrime will Cost Businesses over $2 Trillion by 2019," Retrieved from Juniper Research: https://www. juniperresearch. com/press/pressreleases/cybercrime-cost-businesses-over-2trillion, 2015.

[4] Report. "Report: 2016 saw 8.5 million mobile malware attacks, ransomware and IoT threats on the rise," https://www.techrepublic.com/article/report-2016-saw-8-5-million-mobile-malware-attacks-ransomware-and-iot-threats-on-the-rise/.

[5] J. S. Magdych, T. Rahmanovic, J. R. McDonald, B. E. Tellier, A. C. Osborne, and N. P. Herath, "Secure gateway with firewall and intrusion detection capabilities," Google Patents, 2012.

[6] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and other botnets,” Computer, vol. 50, no. 7, pp. 80-84, 2017.

[7] T. Kothmayr, W. Hu, C. Schmitt, M. Bruenig, and G. Carle, "Poster: Securing the internet of things with DTLS." pp. 345-346.

[8] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, “TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones,” ACM Transactions on Computer Systems (TOCS), vol. 32, no. 2, pp. 5, 2014.

[9] T. Wang, J. Zhou, X. Chen, G. Wang, A. Liu, and Y. Liu, “A Three-Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing,” IEEE Transactions on

Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 3-12, 2018.

[10] A. Altaher, “An improved Android malware detection scheme based on an evolving hybrid neuro-fuzzy classifier (EHNFC) and permission-based features,” Neural Computing and Applications, vol. 28, no. 12, pp. 4147-4157, 2017.

[11] Y. Zhang, J. Pang, F. Yue, and J. Cui, "Fuzzy neural network for malware detect." pp. 780-783.

[12] A. Shalaginov, and K. Franke, “Automatic rule-mining for malware detection employing neuro-fuzzy approach,” Norsk informasjonssikkerhetskonferanse (NISK), vol. 2013, 2013.

[13] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, “Toward credible evaluation of anomaly-based intrusion-detection methods,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 5, pp. 516-524, 2010.

[14] D. Damopoulos, S. A. Menesidou, G. Kambourakis, M. Papadaki, N. Clarke, and S. Gritzalis, “Evaluation of anomaly-based IDS for mobile devices using machine learning classifiers,” Security and Communication Networks, vol. 5, no. 1, pp. 3-14, 2012.

[15] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks." pp. 985-990.

[16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: theory and applications,” Neurocomputing, vol. 70, no. 1-3, pp. 489-501, 2006.

[17] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 42, no. 2, pp. 513-529, 2012.

[18] G.-B. Huang, “What are extreme learning machines? Filling the gap between Frank Rosenblatt’s dream and John von Neumann’s puzzle,” Cognitive Computation, vol. 7, no. 3, pp. 263-278, 2015.

[19] A. Akusok, K.-M. Björk, Y. Miche, and A. Lendasse, “High-performance extreme learning machines: a complete toolbox for big data applications,” IEEE Access, vol. 3, pp. 1011-1025, 2015.

[20] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, “An empirical comparison of botnet detection methods,” computers & security, vol. 45, pp. 100-123, 2014.

[21] F. Afifi, N. B. Anuar, S. Shamshirband, and K.-K. R. Choo, “DyHAP: dynamic hybrid ANFIS-PSO approach for predicting mobile malware,” PloS one, vol. 11, no. 9, pp. e0162627, 2016.

[22] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, “Madam: Effective and efficient behavior-based android malware detection and prevention,” IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 1, pp. 83-97, 2018.

[23] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for android." pp. 15-26.

[24] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, “Evaluation of machine learning classifiers for mobile malware detection,” Soft Computing, vol. 20, no. 1, pp. 343-357, 2016.

[25] M. Sun, X. Li, J. C. Lui, R. T. Ma, and Z. Liang, “Monet: a user-oriented behavior-based malware variants detection system for android,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 5, pp. 1103-1112, 2017.

[26] M. Salehi, and M. Amini, “Android Malware Detection using Markov Chain Model of Application Behaviors in Requesting System Services,” arXiv preprint arXiv:1711.05731, 2017.

[27] A. J. Poulter, S. J. Johnson, and S. J. Cox, “Extensions and Enhancements to “the Secure Remote Update Protocol”,” Future Internet, vol. 9, no. 4, pp. 59, 2017.

[28] G.-B. Huang, “An insight into extreme learning machines: random neurons, random features and kernels,” Cognitive Computation, vol. 6, no. 3, pp. 376-390, 2014.

[29] X. Jiang, and Y. Zhou, "Dissecting android malware: Characterization and evolution." pp. 95-109.

[30] C. E. Shannon, “A mathematical theory of communication,” Bell system technical journal, vol. 27, no. 3, pp. 379-423, 1948.

[31] Z. A. Ahmad Firdaus, “Mobile malware anomaly-based detection systems using static analysis features/Ahmad Firdaus Zainal Abidin,” University of Malaya, 2017.

[32] J. T. Kent, “Information gain and a general measure of correlation,” Biometrika, vol. 70, no. 1, pp. 163-173, 1983.

[33] M. Grill, I. Nikolaev, V. Valeros, and M. Rehak, "Detecting DGA malware using NetFlow." pp. 1304-1309.

[34] S. Roshan, Y. Miche, A. Akusok, and A. Lendasse, “Adaptive and online network intrusion detection system using clustering and Extreme Learning Machines,” Journal of the Franklin Institute, vol. 355, no. 4, pp. 1752-1779, 2018.

[35] D. M. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” 2011.

[36] Q. Gu, Z. Li, and J. Han, “Generalized fisher score for feature selection,” arXiv preprint arXiv:1202.3725, 2012.

[37] CTU. "The CTU-13 dataset a labeled dataset with botnet-normal-and-background-traffic," 27 Feb 2019, 2019; https://mcfp.weebly.com/the-ctu-13-dataset-a-labeled-dataset-with-botnet-normal-and-background-traffic.html#.

Pattern Mining for Knowledge Discovery

Carson K. Leung
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
kleung@cs.umanitoba.ca

ABSTRACT

Pattern mining aims to discover from data the implicit, previously unknown and potentially useful information and knowledge in the form of patterns. Over the past 20 years, numerous pattern mining algorithms have been proposed. They focused on algorithmic efficiency, functionalities, and other aspects. These algorithms have been applied to various real-life applications running in serial, parallel, and/or high-performing computing environments. In this paper, we review many existing pattern mining algorithms and suggest some pattern mining algorithms—especially, hybrid vertical frequent pattern mining running in serial, parallel, high-performing computing, and/or edge/fog environments—to discover knowledge from dataset.

CCS CONCEPTS

- **Information systems ~ Data mining**
- Information systems ~ Data analytics
- Theory of computation ~ Parallel algorithms
- Theory of computation ~ Distributed algorithms
- Theory of computation ~ MapReduce algorithms
- Computing methodologies ~ Distributed computing methodologies
- Computing methodologies ~ MapReduce algorithms

KEYWORDS

Data mining, Frequent pattern mining, Hybrid algorithm, Vertical mining, Item-centric mining, High performance computing, Edge computing, Fog computing

ACM Reference format:

Carson K. Leung. 2019. Pattern Mining for Knowledge Discovery. In *Proceedings of 23rd International Database Engineering & Applications Symposium (IDEAS'19). ACM, New York, NY, USA, 5 pages.* https://doi.org/10.1145/3331076.3331099

IDEAS'19, June 10–12, 2019, Athens, Greece
©2019 Copyright is held by the owner/author. Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6249-8/19/06...$15.00
https://doi.org/10.1145/3331076.3331099

1 Introduction

Data mining [1,2] aims to discover implicit, previously unknown and potentially useful information and knowledge from data. As a popular data mining task, *pattern mining* discovers knowledge in the form of patterns (e.g., discriminative patterns [3], 'following' patterns [4], frequent patterns/itemsets [5,6], frequent subgraphs [7], periodic patterns [8], sequential patterns [9]). For instance, *frequent pattern mining* (aka frequent itemset mining) finds frequently occurring patterns such as sets of frequently co-occurring items or events. Frequent pattern mining can be served as either a stand-alone mining task or a pre-processing step for other data mining tasks (e.g., association rule mining, sequential mining, associative classification). For instance, frequent patterns mined from shopper market transactions reveal baskets of popular merchandise items purchased by customers. Similarly, frequent patterns mined from web logs reveal collections of frequently visited webpages. In addition, the mined frequent patterns can be served as building blocks for other data mining tasks like association rule mining [10], which aims to discover association rules of the form $A \Rightarrow C$ for revealing association relationships between frequent patterns A and C. Similarly, sequential mining aims to find temporally frequent sequences of frequent patterns. Moreover, associative classification aims to classification rules of the form $A \Rightarrow L$ for classifying frequent pattern A with a class label L.

Over the past 20 years, frequent pattern mining has drawn attention of many researchers, who developed numerous frequent pattern mining algorithms. Many of the early frequent pattern mining ones were *serial* algorithms—such as Apriori-based [11], which depend on a generate-and-test paradigm to mine frequent patterns from transaction datasets by first generating candidates and then checking their actual frequency (i.e., occurrences) against the dataset. To improve algorithmic efficiency, other serial frequent pattern mining algorithms—such as tree-based frequent pattern mining algorithms (e.g., FP-growth [12]), hyperlinked array based frequent pattern mining algorithms (e.g., H-mine [13]), and bitwise-based frequent pattern mining algorithms (e.g., B-mine [14])—have been developed. Besides the transaction-centric algorithms that mine the datasets "horizontally", there are also item-centric algorithms—such as Eclat [15], dEclat [16], and VIPER [17]—that mine the datasets "vertically". Depending on factors like data density, one of these item-centric frequent pattern mining algorithm can run faster than another.

Besides the aforementioned serial algorithms for the discovery of frequent patterns, there are also *distributed and parallel* frequent pattern mining algorithms [18-21]. They use multiple processors that either (a) have access to a *shared memory* with which information is exchanged among processors via Open Multi-Processing (OpenMP) or (b) have their own *private memory* (i.e., distributed memory) with which for information is exchanged (i.e., messages are passed) among processors via Message Passing Interface (MPI). These processors are usually in a *computer cluster* (which is a group of distributed or parallel computers that are interconnected through high-speed local area networks (LAN) or worked together as a single computing unit to mine the data) or a *computer grid* (which is a loosely coupled group of coordinated heterogeneous networked computers—in which each computer may perform a different mining process) [22].

Advancements in technology enable easy generation and collection of huge volumes of valuable data (which may be of different veracity level) at a high velocity from a wide variety of data sources such as Internet of Things (IoT) devices. *Data science* is in demand for (a) discovering knowledge from these big data, and for (b) visualizing, validating and interpreting these big data and the mined results (i.e., knowledge discovered from these big data). In recent years, frequent pattern mining algorithms use the concept of MapReduce programming model [23] implemented in the Apache Hadoop or Apache Spark framework [24] to mine frequent patterns in a *cloud computing* environment [25]. Here, as implied by its name, MapReduce uses (a) the map function to transform each value in an input list to a mapped value in an output list and (b) the reduce function to combine values in an input list to a single reduced value as an output. By do so, the data miner only need to focus on specifying the map and reduce functions—without worrying about implementation details on how to handle machine failures, manage inter-machine communication, partition the input data, and/or schedule and execute the program across multiple machines. Apache Hadoop relies on the Hadoop Distributed File System (HDFS) to store data on commodity machines and provide high aggregate bandwidth across the computer cluster. In contrast, Apache Spark relies on a read-only multiset of data items—called resilient distributed dataset (RDD)—distributed over a cluster of machines and maintained in a fault-tolerant fashion. Cloud computing utilizes a network of remote servers that are hosted on the Internet—such as cloud data center or cloud computing services like Amazon Web Services (AWS)—for the storage, management, processing, and analysis of data. Different types of clouds (e.g., public, private, hybrid clouds) involve groups of interconnected and virtualized computers to provide on-demand services such as mining-as-a-service (MaaS) [26]. While efficient, many of these high performance computing (HPC) based frequent pattern mining algorithms transmit data to clouds for mining. For example, MREclat [27] and Dist-Eclat [28] are MapReduce versions of Eclat and dEclat, respectively, whereas BigFIM [28] is a MapReduce version of a hybrid of the Apriori and dEclat algorithms.

To reduce the amount of data transmission and thus to speed up the mining process, frequent pattern mining can be performed by using *edge computing* aka *fog computing* [29-31]. The idea is to store, manage, process and analyze data in the fog or on the edge devices. In other words, the computation is performed on local IoT devices or the networking services that are located in between the local IoT devices and the data cloud centers. In this paper, we focus on vertical mining (i.e., item-centric frequent pattern mining). Recall that the runtimes of item-centric frequent pattern mining algorithms vary depending of factors like data density. Our *first key contribution* of this paper is hybrid algorithms, which switch among Eclat [15], dEclat [16] and VIPER [17] algorithms in order to take the benefits of the three worlds. Our *second key contribution* of this paper is edge/fog computing based vertical mining algorithms, which reduce the amount of data transmission and thus speeds up the mining process.

The reminder of this paper is organized as follows. The next section gives background and related works. Section 3 presents our algorithms. Section 4 briefly discusses the evaluation results of our feasibility studies. Finally, conclusions are drawn in Section 5.

2 Background and Related Works

The classical Apriori algorithm [11] applies a generate-and-test paradigm in mining frequent patterns in a level-wise bottom-up fashion. Specifically, the algorithm first generates candidate patterns of cardinality k (i.e., candidate k-itemset) and then tests if each of them is frequent (i.e., tests if its frequency meets or exceeds the user-specified minimum frequency threshold). Based on these frequent patterns of cardinality k (i.e., frequent k-itemsets), the algorithm then generates candidate patterns of cardinality k+1 (i.e., candidate (k+1)-itemsets). This process is applied repeatedly to discover frequent patterns of all cardinalities. A disadvantage of the Apriori algorithm is that it requires $K_{\max}$ scans of the database to discover all frequent patterns (where $K_{\max}$ is the maximum cardinality of discovered patterns). During each database scan, the algorithm mines frequent patterns in a *transaction-centric* fashion that it finds what k-itemset is supported by (or contained in) a transaction.

Like the classical Apriori algorithm, the Eclat algorithm [15] also uses a level-wise bottom-up paradigm to mine frequent patterns. However, it does so by using an *item-centric* fashion that it counts the number of transactions supporting or containing the patterns. To elaborate, with Eclat, the database is treated as a collection of item lists. Each list for an item x keeps IDs of transactions containing x. The length of the list for x gives the frequency of 1-itemset $\{x\}$. By taking the intersection of lists for two frequent itemsets α and β, we get the IDs of transactions containing (α∪β). Again, the length of the resulting (intersected) list gives the frequency of the pattern (α∪β). Eclat works well when the database is sparse. However, when the database is dense, these item lists can be long.

As an extension to Eclat, the dEclat algorithm [16] also uses a level-wise bottom-up paradigm. Unlike Eclat (which uses keeps sets of IDs of transactions containing itemsets, i.e., tidsets), dEclat uses diffset which is the set difference between tidsets of two related itemsets. Specifically, the diffset of a k-itemset $\alpha = \gamma\cup\{a\}$, where γ is its (k–1)-prefix, is defined as the difference between the tidset of α and the tidset of γ. To start the mining process, dEclat computes the diffset of 1-itemset $\{x\}$ by taking the complement of the tidset of $\{x\}$, i.e.,

$$\text{diffset}(\{x\}) = \text{tidset}(\text{TDB}) - \text{tidset}(\{x\}) = \{t_j \mid x \notin t_j \subseteq \text{TDB}\},$$

where t_j represents the j-th transactions in the transaction database TDB. The diffset of $\{x\}$ then captures those transactions that do not contain $\{x\}$. For a transaction database TDB containing n transactions, the frequency of 1-itemset $\{x\}$ can then be computed by

$$n - |\text{diffset}(\{x\})|,$$

where $|\text{diffset}(\{x\})|$ represents the length of diffset($\{x\}$). Afterwards, let k-itemsets $\alpha = \gamma \cup \{a\}$ and $\beta = \gamma \cup \{b\}$ sharing a common (k–1)-prefix γ such that $(\alpha \cup \beta) = \gamma \cup \{a, b\}$. Then, taking the set difference between diffset(β) and diffset(α) gives the diffset of the resulting ($\alpha\cup\beta$). The frequency of ($\alpha\cup\beta$) can be computed by subtracting the length of diffset($\alpha\cup\beta$) from the frequency of α. dEclat works well when the database is dense. However, when the database is sparse, these diffsets can be long.

Alternatively, the VIPER algorithm [17] represents the item lists in the form of bit vectors. Each bit in a vector for a domain item x indicates the presence (bit "1") or absence (bit "0") of transaction containing x. The number of "1" bits for x gives the support of 1-itemset $\{x\}$. By computing the dot product of vectors for two frequent itemsets α and β, we get the vector indicating the presence of transactions containing ($\alpha \cup \beta$). Again, the number of "1" bits of this vector gives the frequency of the resulting pattern ($\alpha \cup \beta$). VIPER works well when the database is dense. However, when the database is sparse, lots of space may be wasted because the vector contains lots of 0s.

3 Our Hybrid Vertical Mining Algorithms

3.1 Our Serial Hybrid Vertical Mining Algorithm

In Section 2, we described three key item-centric serial frequent pattern mining algorithms—namely, Eclat, dEclat, VIPER. Depending on the data density, one algorithm would perform better than others. In other words, there is no clear winner. As such, we suggest here a *serial* hybrid algorithm that takes the best of the three worlds.

As frequent patterns are mined using a level-wise bottom-up paradigm in all these three algorithms, our hybrid algorithm switches from one to another based on the densities of dataset: (a) If the dataset is dense, our hybrid algorithm switches from using transaction IDs to using diffsets early, i.e., switching from Eclat to dEclat early. (b) If the dataset is sparse, our hybrid algorithm uses transaction IDs for longer period of mining time before it switches to diffsets, i.e., switching from Eclat to dEclat late. (c) If the number of transaction IDs in tidsets (for Eclat) or diffsets (for dEclat) higher than ~12.5% of the product of the numbers of all transactions and domain items, our hybrid algorithm uses bit vectors than transaction IDs, i.e., switching from Eclat or dEclat to VIPER.

3.2 Our HPC Based Hybrid Vertical Mining Algorithm with MapReduce

To handle big data, we extend our serial item-centric hybrid algorithm to become a *high performance computing* (*HPC*) *based* item-centric hybrid algorithm. Specifically, we distribute the original horizontal transaction-centric dataset and to store the transactions as frequent equivalence classes in different units (i.e., transformed into a vertical item-centric dataset). With the map function, the mappers apply hybrid vertical mining on each partition without the need of any additional information from other workers. Unlike the traditional vertical mining algorithms like Eclat or dEclat, our hybrid algorithm does not choose just a single strategy. Instead, it chooses different strategies based on the densities of datasets. Specifically, our hybrid algorithm first captures transaction IDs (i.e., tidsets), which consumes less time in calculating the support. Our hybrid algorithm then computes differences among the sets of transaction IDs (i.e., diffsets). The switching from one strategy to another is based on the densities of datasets as stated in Section 3.1.

3.3 Our Edge/Fog Computing Based Hybrid Vertical Mining Algorithm by Using Networking Services

When mining patterns with cloud computing, local data are transmitted from IoT devices to *data cloud centers*, in which data are then partitioned and redistributed among several worker nodes. To mine patterns from IoT devices with edge/fog computing, local data are transmitted from the IoT devices to their *local networking services* lying in between the IoT devices and the usual data centers. Afterwards, the local networking services then perform data aggregation and data mining in order to discover locally frequent patterns (i.e., patterns that are locally frequent with respect to the data collected from one or more IoT devices served by the local networking services). Here, depending on the representation of transaction IDs (e.g., whether data are represented by tidsets, diffsets or bit vectors), our algorithm applies set intersections or dot products of represented data to find patterns that are locally frequent on the data supported by the networking services. Frequency of these patterns can be computed by measuring the size/length of tidsets or diffsets or by counting the number of 1s in the bit vectors. Once the data representation switches, our hybrid algorithm uses the corresponding mining techniques (e.g., intersection, dot product).

Then, our hybrid algorithm takes the union of these locally frequent patterns. After taking the union to form global candidate patterns, the algorithm calculates the frequency of these global candidates on the local IoT device. The algorithm then transmits the frequency values of these candidates to other IoT devices (or a master node), and sums the frequency values in order to discover globally frequent patterns in the network.

3.4 Our Edge/Fog Computing Based Hybrid Vertical Mining Algorithm on Local IoT Devices

When mining patterns with cloud computing, local data are transmitted from IoT devices to *data centers*, in which data are then partitioned and redistributed among several worker nodes. When mining patterns with edge/fog computing through local network services, local data are transmitted from IoT devices to *local networking services*. Knowing that some IoT devices are more powerful in such a way that more computations can be performed locally, we explore an alternative algorithm for mining patterns with edge/fog computing. Specifically, to mine patterns from IoT devices with edge/fog computing, local data are kept on individual IoT devices in order to discover locally frequent patterns (i.e., patterns that are locally frequent with respect to the data collected from individual IoT devices). Again, depending on the representation of transaction IDs (e.g., whether data are represented by tidsets, diffsets or bit vectors), our algorithm applies set intersections or dot products of represented data to find patterns that are locally frequent on the data supported by the networking services. Frequency of these patterns can be computed by measuring the size/length of tidsets or diffsets or by counting the number of 1s in the bit vectors. Once the data representation switches, our hybrid algorithm uses the corresponding mining techniques (e.g., intersection, dot product).

After taking the union of these locally frequent patterns to form global candidate patterns, we calculate the frequency of these global candidates on each IoT device. We then transmit their frequency values to other IoT devices (or a master node), and sum the frequency values in order to discover globally frequent patterns in the network.

4 Evaluation Results

To evaluate our hybrid frequent pattern mining algorithm, we first set up a feasibility study. We then set up experiments to compare our algorithm with existing frequent pattern mining algorithms by using benchmark datasets.

Results of our feasibility study on our serial hybrid algorithm suggests that it switches from using tidsets to using diffsets when the frequency of the subset is at least half of that of the superset. Similar results apply to our three other hybrid algorithms.

Results of our feasibility study on our MapReduce based hybrid algorithm suggests that it leads to a benefit of the switch. Specifically, as each worker performs the vertical mining simultaneously, each worker may choose a different strategy based on the current system load. Moreover, as another benefit, our hybrid algorithm only needs to scan the database once in the entire mining process. Once vertical mining is performed by each worker, the results (i.e., frequent itemsets) are collected from these workers to the driver.

Results of our feasibility study on mining frequent patterns with edge/fog computing through local networking services show that it is more efficient and practical than apply the hybrid algorithm on cloud computing. A reason is that, when compared with cloud-computing based mining, computations for mining and analysis with edge/fog computing based mining are performed closer to end-users.

Results of our feasibility study on mining frequent patterns with edge/fog computing on local IoT devices show that it is more efficient and practical than that with edge/fog computing through local networking services, which was shown to be more efficient and practical than that with cloud computing. A reason is that, when compared with cloud-computing based mining, computations for mining and analysis with edge/fog computing based mining on local IoT devices are performed closer to end-users. Computations for mining and analysis with edge/fog computing based mining through local network services are performed even much closer to end-users.

Pattern mining with these two edge/fog computing approaches reduces the latency and network bandwidth, enables geographic focus, and increases reliability and security.

5 Conclusions

Pattern mining aims to discover from data the implicit, previously unknown and potentially useful information and knowledge in the form of patterns. As a popular pattern mining task in the data mining domain, numerous frequent pattern mining algorithms have been proposed over the past 20 years. They focused on algorithmic efficiency, functionalities, and other aspects. These algorithms have been applied to various real-life applications running in serial, parallel, and/or high-performing computing environments. Examples include the classical Apriori algorithm, which is a transaction-centric level-wise algorithm that mines frequent patterns in a bottom-up fashion. Other examples include the Eclat, dEclat and VIPER algorithms, which are item-centric level-wise algorithms that also mine frequent patterns in a bottom-up fashion.

In this paper, we reviewed many existing frequent pattern mining algorithms including the aforementioned serial algorithms, as well as other distributed and parallel algorithms, and algorithms running in high performance computing (HPC) environments (e.g., computer clusters, computer grids, data cloud centers). Our key contributions include our presentation of item-centric level-wise hybrid algorithms that mine frequent patterns in serial, in the MapReduce environment, using networking services, and on local IoT devices. These hybrid algorithms take the benefits of all worlds, in which data are represented as collections of transaction IDs (as in Eclat), differences/changes among collections of transaction IDs (as in dEclat), and collections of bit vectors (as in VIPER). The use of edge/fog computing in vertical frequent pattern mining enables the mining to be performed closer to the end users, the reduction of transmitted data, and the saving of runtime. As ongoing and future work, we further explore other performance enhancements.

ACKNOWLEDGMENTS

This work is partially supported by NSERC (Canada) and the University of Manitoba.

REFERENCES

[1] W.J. Frawley, G. Piatetsky-Shapiro, & C.J. Matheus, Knowledge discovery in databases: an overview, AI Magazine, 13(3), pp. 57-70, 1992.

[2] S. Daggumati & P.Z. Revesz, Data mining ancient script image data using convolutional neural networks, IDEAS 2018, pp. 267-272.

[3] K. Vaculík & L. Popelínsky, WalDis: mining discriminative patterns within dynamic graphs, IDEAS 2018, pp. 95-102.

[4] C.K. Leung, R. Middleton, A.G.M. Pazdor, & Y. Won, Mining 'following' patterns from big but sparsely distributed social network data, IEEE/ACM ASONAM 2018, pp. 916-919.

[5] N. Abuzayed & B. Ergenç, Comparison of dynamic itemset mining algorithms for multiple support thresholds, IDEAS 2017, pp. 309-316.

[6] C.K. Leung, C.S.H. Hoi, A.G.M. Pazdor, B.H. Wodi, & A. Cuzzocrea, Privacy-preserving frequent pattern mining from big uncertain data, IEEE BigData 2018, pp. 5101-5110.

[7] M.A. Islam, C.F. Ahmed, C.K. Leung, & C.S.H. Hoi, WFSM-MaxPWS: an efficient approach for mining weighted frequent subgraphs from edge-weighted graph databases, PAKDD 2018, Part III, pp. 664-676.

[8] A.K. Chanda, C.F. Ahmed, M. Samiullah, & C.K. Leung, A new framework for mining weighted periodic patterns in time series databases, ESWA 79, pp. 207-224, 2017.

[9] M.M. Rahman, C.F. Ahmed, & C.K. Leung, Mining weighted frequent sequences in uncertain databases, Information Sciences 479, pp. 76-100, 2019.

[10] C.K. Leung, F. Jiang, & A.G.M. Pazdor, Bitwise parallel association rule mining for web page recommendation, IEEE/WIC/ACM WI 2017, pp. 662-669.

[11] R. Agrawal & R. Srikant, Fast algorithms for mining association rules in large databases, VLDB 1994, pp. 487-499.

[12] J. Han, J. Pei, & Y. Yin, Mining frequent patterns without candidate generation, ACM SIGMOD 2000, pp. 1-12.

[13] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, & D. Yang, H-mine: hyper-structure mining of frequent patterns in large databases, IEEE ICDM 2001, pp. 441-448.

[14] F. Jiang, C.K. Leung, & H. Zhang, B-mine: frequent pattern mining and its application to knowledge discovery from social networks, APWeb 2016, Part I, pp. 316-328.

[15] M.J. Zaki, Scalable algorithms for association mining, IEEE TKDE, 12(3), pp. 372- 390, 2000.

[16] M.J. Zaki, Fast vertical mining using diffsets, ACM KDD 2003, pp. 326-335.

[17] P. Shenoy, J.R. Bhalotia, M. Bawa, & D. Shah, Turbo-charging vertical mining of large databases, ACM SIGMOD 2000, pp. 22-33.

[18] C.K. Leung, F. Jiang, & A.G.M. Pazdor, Bitwise parallel association rule mining for web page recommendation, IEEE/WIC/ACM WI 2017, pp. 662-669.

[19] C.K. Leung & H. Zhang, Management of distributed big data for social networks, IEEE/ACM CCGrid 2016, pp. 639-648.

[20] S.K. Tanbeer, C.F. Ahmed, & B. Jeong, Parallel and distributed frequent pattern mining in large databases, IEEE HPCC 2009, pp. 407-414.

[21] M.J. Zaki, Parallel and distributed association mining: a survey, IEEE Concurrency, 7(4), pp. 14–25, 1999.

[22] T.T.Q. Nguyen, C. Bobineau, V. Debusschere, Q. Giap, & N. Hadj-Said, Using declarative programming for network data management in smart grids, IDEAS 2018, pp. 292-296.

[23] P. Braun, A. Cuzzocrea, F. Jiang, C.K. Leung, & A.G.M. Pazdor, MapReduce-based complex big data analytics over uncertain and imprecise social networks, DaWaK 2017, pp. 130-145.

[24] O.A. Sarumi, C.K. Leung, & A.O. Adetunmbi, Spark-based data analytics of sequence motifs in large omics data, Procedia Computer Science 126, pp. 596-605, 2018.

[25] O. Kamel, A. Chaoui, & M. Gharzouli, Cloud service composition modeling using bigraphical reactive systems, IDEAS 2017, pp. 40-48.

[26] Z. Han & C.K. Leung, FIMaaS: scalable frequent itemset mining-as-a-service on cloud for non-expert miners, BigDAS 2015, pp. 84-91.

[27] Z. Zhang, G. Ji, & M. Tang, MREclat: an algorithm for parallel mining frequent itemsets,. CBD 2013, pp. 177-180.

[28] M. Snady, A. Emin, & G. Bart, Frequent itemset mining for big data, IEEE BigData 2013, pp. 111-118.

[29] P. Braun, A. Cuzzocrea, C.K. Leung, A.G.M. Pazdor, J. Souza, & S.K. Tanbeer, Pattern mining from big IoT data with fog computing: models, issues, and research perspectives, IEEE/ACM CCGrid 2019, pp. 854-891.

[30] V. Dastjerdi & R. Buyya, Fog computing: helping the Internet of Things realize its potential, IEEE Computer 49(8), pp. 112-116, 2016.

[31] C.K. Leung, D. Deng, C.S.H. Hoi, & W. Lee, Constrained big data mining in an edge computing environment, BigDAS 2017, pp. 61-68.

The Design and Implementation of AIDA: Ancient Inscription Database and Analytics System

Peter Z. Revesz
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
revesz@cse.unl.edu

M. Parvez Rashid
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
parvezaiub@gmail.com

Yves Tuyishime
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
ytuyishime@unl.edu

ABSTRACT

This paper describes the development of AIDA, the Ancient Inscription Database and Analytics system. The AIDA system currently stores three types of ancient Minoan inscriptions: Linear A, Cretan Hieroglyph and Phaistos Disk inscriptions. In addition, AIDA provides candidate syllabic values and translations of Minoan words and inscriptions into English. The AIDA system allows the users to change these candidate phonetic assignments to the Linear A, Cretan Hieroglyph and Phaistos symbols. Hence the AIDA system provides for various scholars not only a convenient online resource to browse Minoan inscriptions but also provides an analysis tool to explore various options of phonetic assignments and their implications. Such explorations can aid in the decipherment of Minoan inscriptions.

CCS CONCEPTS

• **Information systems → Database design and models**; **Digital libraries and archives**; **Dictionaries**; *Data mining*; *Document representation*; *Specialized information retrieval.*

KEYWORDS

Cretan Hieroglyphs, Data Analytics, Data Mining, Database Design, Linear A, Minoan, User Interface

ACM Reference Format:
Peter Z. Revesz, M. Parvez Rashid, and Yves Tuyishime. 2019. The Design and Implementation of AIDA: Ancient Inscription Database and Analytics System. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3331076.3331117

1 INTRODUCTION

The ancient Bronze Age Minoan culture flourished on the island of Crete and some other islands and coastal areas of the Aegean Sea between about 3000 and 1500 BCE [10]. The Minoan language, a Pre-Greek, non-Indo-European language, survives only in ancient inscription in three different types of scripts, namely the Linear A script (about 1500 inscriptions), the Cretan Hieroglyphic script (about 350 inscriptions), and the Phaistos Disk inscription, which has a unique inscription consisting of printed seals for each symbol [11]. There are no widely accepted decipherments of these three Minoan inscriptions, although there are many proposals. One problem with decipherment attempts is that there are too few longer inscriptions in the three different scripts. About 1200 Linear A inscriptions contain only one or two symbols. It would be highly beneficial for a decipherment effort to bring together all three types of inscriptions into a common format. Since Linear A inscriptions are the most common, this would mean in practice the translation of the Cretan Hieroglyph and the Phaistos Disk inscriptions into Linear A. That is one of the goals of our Minoan database system. The basis of the translation to Linear A are two functions. First, a mapping from the Cretan Hieroglyph symbols to the Linear A symbols. Second a mapping from the Phaistos symbols to Cretan Hieroglyph symbols.

We present the AIDA system, short for Ancient Inscription Database and Analytics system, which brings all three types of Minoan inscriptions into the same Linear A format and provides a powerful search capability. The acronym name AIDA is famous from Verdi's opera of the same name, where the Ethiopian princess is called Aida. That name is said to derive from Aita, an ancient Egyptian or other African women's name. It may be also cognate with Finnish äita, which means "mother" in English. In any case, one of the major goals of the AIDA system is to find possible cognates of the Minoan words.

In AIDA, one can enter any Linear A sequence and all the words and the database system will return all the words and inscriptions that contain that sequence including the Cretan Hieroglyph inscriptions and Phaistos Disk blocks whose translations into Linear A contain the search sequence. Similarly, one can search a Cretan Hieroglyph sequence and bring up all three types of inscriptions that contain the equivalent signs. In addition, our system provides the English meaning of a set of words from the lexicon in [16] and translations of texts from [14–17].

The rest of this paper is organized as follows. Section 2 describes all the data sources for our research. Section 3 shows the entity relationship diagram of our database and outlines the main implementation features. Section 4 describes the AIDA system's user interface and some queries. Section 5 outlines the data analytics that the AIDA system is planned to perform. Section 6 discusses related work. Finally, Section 7 gives some conclusions and directions for further research.

2 DATA SOURCES

For the Cretan Hieroglyphic inscriptions we used the book *Corpus Hieroglyphicarum Inscriptionum Cretae*, abbreviated CHIC, by Olivier et al. [12]. For the Linear A inscriptions we used Godart and

IDEAS'19, June 10-12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331117

Olivier's book *Recueil des inscriptions en Linéaire A* [8], which is commonly abbreviated GORILA by the first letters of the authors and the title. For the Phaistos Disk, we used Evans [7]. These three reference books introduced, respectively, a special numbering of the Cretan Hieroglyph, Linear A and Phaistos Disk symbols. The CHIC book also gave a numbering of the Cretan Hieroglyphic inscriptions. Evans [7] called the two sides of the Phaistos Disk, sides A and B and gave a numbering of the blocks on side A from A1 in the inside to A30 on the outside and on side B from B1 in the inside to B31 on the outside.

Cretan Hieroglyphs and Linear A are just two of about ten different scripts that belong to the *Cretan Script Family*, whose development was studied using bioinformatics phylogenetic algorithms in Revesz [13]. The discovery of the Cretan Script Family played an essential role in the decipherment of the Phaistos Disk [15], Cretan Hieroglyphic inscriptions [17] and Linear A [16]. All these decipherments were based on one-to-one mappings between pairs of scripts within the Cretan Script Family. When a script with known phonetic values is mapped to a script with unknown phonetic values, then the phonetic values of the former script also can be mapped, at least tentatively, to the symbols of the latter script.

Revesz also gave one-to-one mappings from the Phaistos Disk symbols and to the Cretan Hieroglyphs [13] and from the Cretan Hieroglyphs to the Linear A symbols [17]. These mappings enable the transliteration from any of the three types of Minoan inscriptions into the other two types.

3 DATABASE DESIGN AND IMPLEMENTATION

Our entity-relationship diagram is shown in Figure 1. The entity relationship diagram contains a relation for the Phaistos Disk symbols (PD-Symbol), the Cretan Hieroglyph symbols (CH-Symbol) and the Linear A symbols (LA-Symbol). These three sets of symbols are indexed, respectively, by the identification numbers given by Evans [7], CHIC [12], and GORILA [8]. We also have relations that store the Phaistos Disk block numerical sequences (PD-Block), the Cretan Hieroglyph number sequences (CH-Inscriptions), and the Linear A words (Lin-A-Lexicon). Between any type of inscriptions and the corresponding type of symbols, there is a many-to-many containment relation. Therefore, there are three containment relations: Contains-PD, Contains-CH and Contains-LA. Finally, relation Lin-A-inscriptions stores the translated Linear A inscriptions by a number sequence and a meaning. There is a many-to-many relationship between the Lin-A-Lexicon relation and the Lin-A-Inscriptions relation. For each Lin-A-Lexicon tuple we store the Linear A word's number sequence as well as its meaning, which is an English word or phrase. We indicate one-to-one relationships by arrows and the number 1 on the links between the entity sets and the relationship set. Similarly, we also indicate by the symbols *N* and *M* on the links the many-to-many relationships.

For the implementation, we used the MYSQL database system for storing and retrieving data. We built the system interface, which will be described in more detail in Section 4.1, using Boostrap V4.3.1, HTML and CSS. We are running a PHP script to handle the input from the user interface and provide output to the users. The whole system is hosted at the following University of Nebraska-Lincoln server: *https://cse.unl.edu/~revesz/aida.php*.

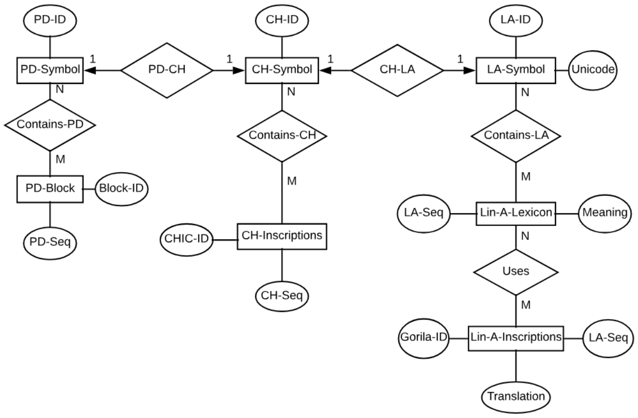


Figure 1: The entity-relationship diagram.

4 THE USER INTERFACE AND QUERIES

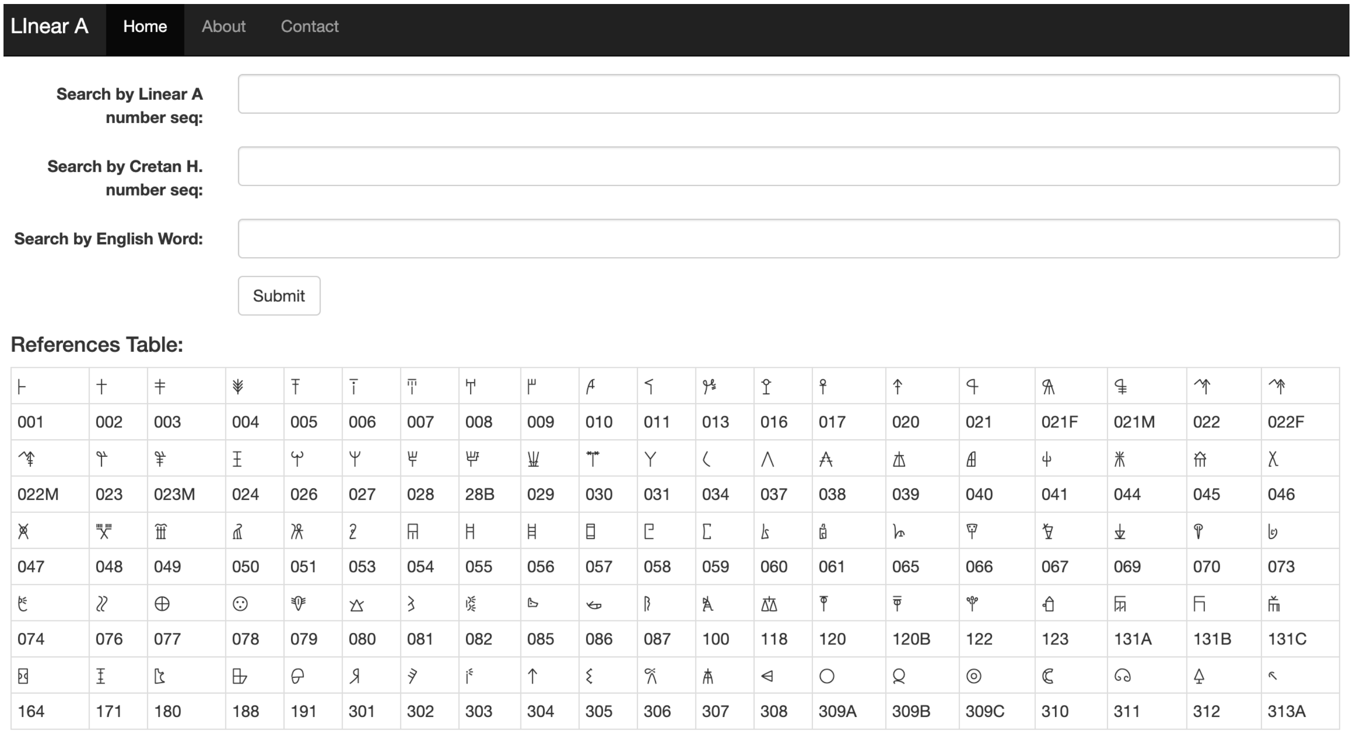
Next we describe the AIDA system's user interface in Section 4.1. After that, the following three sections present different types of queries. In particular, Section 4.2 presents Linear A queries, Section 4.3 presents Cretan Hieroglyph queries, and Section 4.4 presents English word queries.

4.1 The User Interface

Figure 2 shows the user interface of the AIDA system. The top line of the user interface contains some clickable choices regarding various information options about the AIDA system, including a brief user's manual that describes how to use the system. The next three lines of the user interface shows three prompt boxes. The user can select any of these three prompt boxes to enter a query. The first prompt box allows the user to enter a Linear A number sequence. The second prompt box allows the user to enter a Cretan Hieroglyph number sequence. The third prompt box allows the user to enter an English keyword. In case the user knows the actual symbol sequence but forgot the associated numbers, the bottom of the AIDA user interface shows a matrix of Linear A symbols. Below each Linear A symbol, its identification number is given based on the GORILA book [8].

4.2 Linear A Queries

By Linear A queries we mean queries that search for the occurrences of various substrings in the Minoan lexicon and the Minoan inscriptions stored in the AIDA system. As an example of a Linear A query, we use the sequence 57-7-67. Given that number sequence, the system returns the answer shown in Figure 3. We see that it is used in three different Linear A inscriptions. For these inscriptions the entire Linear A number sequences and the GORILA identification strings are returned. After the GORILA identification string we also list in parentheses the GORILA volume number and page number separated by a slash where the inscription is described.



Search by Linear A number seq:

Search by Cretan H. number seq:

Search by English Word:

References Table:

001	002	003	004	005	006	007	008	009	010	011	013	016	017	020	021	021F	021M	022	022F
022M	023	023M	024	026	027	028	28B	029	030	031	034	037	038	039	040	041	044	045	046
047	048	049	050	051	053	054	055	056	057	058	059	060	061	065	066	067	069	070	073
074	076	077	078	079	080	081	082	085	086	087	100	118	120	120B	122	123	131A	131B	131C
164	171	180	188	191	301	302	303	304	305	306	307	308	309A	309B	309C	310	311	312	313A

Figure 2: The AIDA user interface.

In addition, the sequence 57-7-67 also occurs in several Linear A lexicon words. One of the lexicon words means "star" while other lexicon words mean "moon." It appears that in the Minoan language the word for "moon" is expressed as either the compound "star+queen" or "star+head", that is, the moon was viewed as the queen or the chief of the stars.

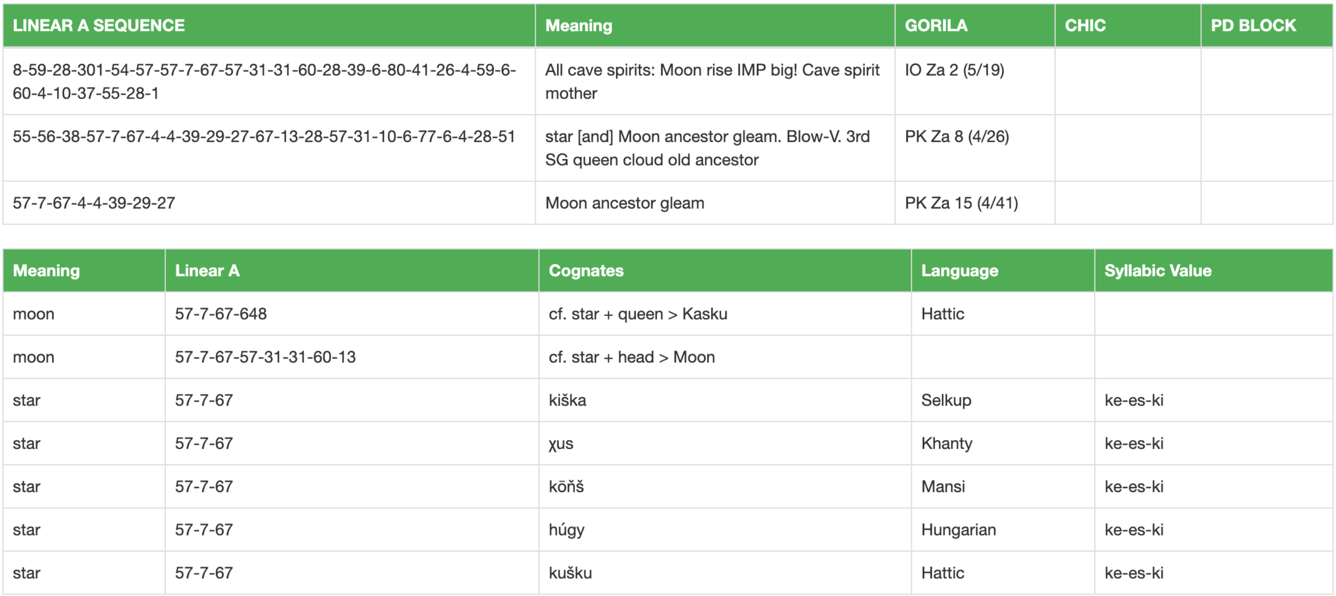
The AIDA system also returns in the last column the syllabic transliteration of the Linear A word for star. The syllabic values are based on Table 12 in [16]. Figure 3 shows that the syllabic value for "star" is *ke-es-ki*. The syllabic values of the Linear A symbols can be updated by the users, which would allow some experimentation. However, any change of syllabic value of a symbol needs to be carefully investigated for its implications. The AIDA system is designed to facilitate such an investigation because the users can retrieve all the words and previous translations that may contain a particular symbol and then see the effect of any change.

The AIDA system also displays in the third and fourth column the putative cognates and the languages in which those cognates occur, respectively. For example, the word *kiška* is a Selkup word that also means "star" in that language. Note the phonetic similarity between *ke-es-ki*, which was likely pronounced as *keski* and the Selkup word *kiška*. The phonetic similarities and the same meaning suggest that they are cognate words. Other possible cognate words retrieved by the AIDA system are *χus* in Khanty, *kō̆ńs* in Mansi and *kušku* in Hattic, all meaning "star."

4.3 Cretan Hieroglyph Queries

Similar to Linear A queries, a Cretan Hieroglyph query retrieves all the Minoan inscriptions that contain a particular Cretan Hieroglyph sequence of its Phaistos Disk or Linear A equivalent sequences. As an example of a Cretan Hieroglyph query, we used the sequence 25-04-03 as shown in Figure 4.

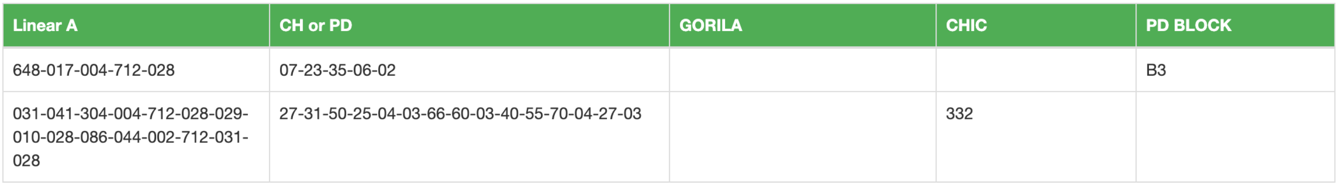
The AIDA system gave an output table where the first column shows the equivalent Linear A sequences of two Minoan inscriptions. The first inscription is a block of the Phaistos Disk, namely block B3. Normally under the CHIC column we would have the Cretan Hieroglyphic inscription identification number from [12], which ranges from #1 to #331. However, there are a few inscriptions that can be considered Cretan Hieroglyph inscriptions, although they do not appear in [12]. One of these inscription is the Arkalochori Axe inscription, which we added to the database as the Cretan Hieroglyphic inscription CHIC #332. The AIDA system was able to bring these two inscriptions with different scripts together and show their relationship. The existence of the common subsequence, which in Linear A would be the following number sequence: 004-712-028, according to the numbering of the Linear A symbols in [8]. The common subsequence implies that it is likely some suffix when the inscriptions are both read from left to right. In a similar manner, a user may find all the occurrences of other candidate prefixes and suffixes. The prefix or suffix nature of the sequences would be strongly supported by their multiple occurrences at the beginning or the end of short inscriptions or the blocks within larger inscriptions such as the Phaistos Disk.



LINEAR A SEQUENCE	Meaning	GORILA	CHIC	PD BLOCK
8-59-28-301-54-57-57-7-67-57-31-31-60-28-39-6-80-41-26-4-59-6-60-4-10-37-55-28-1	All cave spirits: Moon rise IMP big! Cave spirit mother	IO Za 2 (5/19)		
55-56-38-57-7-67-4-4-39-29-27-67-13-28-57-31-10-6-77-6-4-28-51	star [and] Moon ancestor gleam. Blow-V. 3rd SG queen cloud old ancestor	PK Za 8 (4/26)		
57-7-67-4-4-39-29-27	Moon ancestor gleam	PK Za 15 (4/41)		

Meaning	Linear A	Cognates	Language	Syllabic Value
moon	57-7-67-648	cf. star + queen > Kasku	Hattic	
moon	57-7-67-57-31-31-60-13	cf. star + head > Moon		
star	57-7-67	kiška	Selkup	ke-es-ki
star	57-7-67	χus	Khanty	ke-es-ki
star	57-7-67	kōńš	Mansi	ke-es-ki
star	57-7-67	húgy	Hungarian	ke-es-ki
star	57-7-67	kušku	Hattic	ke-es-ki

Figure 3: The result of querying the Linear A sequence 57-7-67.



Linear A	CH or PD	GORILA	CHIC	PD BLOCK
648-017-004-712-028	07-23-35-06-02			B3
031-041-304-004-712-028-029-010-028-086-044-002-712-031-028	27-31-50-25-04-03-66-60-03-40-55-70-04-27-03		332	

Figure 4: The result of querying the Cretan Hieroglyph sequence 25-04-03.

4.4 Word Queries

A word query simply retrieves all the lexicon items and translated texts where some English language keyword appears. The English language keyword can be any word in the English language. If it is not found in the lexicon or the translations, then the AIDA system returns the message "not found." As an example of a word query, we used AIDA to look up all the items that contain the word "light" as shown in Figure 5 and the word "moon" as shown in Figure 6.

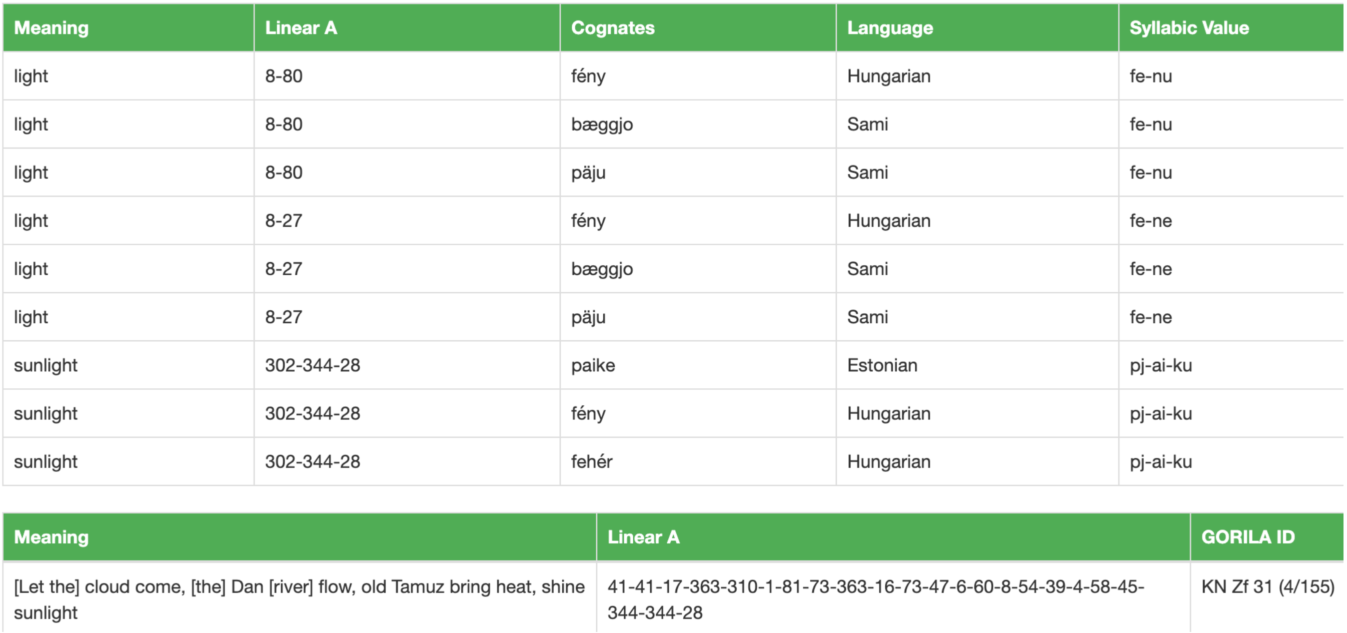
As Figure 5 shows, the word "light" occurs not only in the dictionary entry for "light" but also in the dictionary entry for "sunlight." The entry for "light" is associated with two different Linear A number sequences, the first is 8-27 and the second is 8-80, which has syllabic transliterations fe-ne and fe-nu, respectively. These two pronunciations may have been dialectical variations, or they may had slightly different connotations that currently we do not know. However, both of these words seem cognate with other words such as *fény* in Hungarian and *päju* in Sami.

The word for "sunlight" has the Linear A number sequence 302-344-28, syllabic transliteration pj-ai-ku and possible cognate *paike* in the Estonian language, where the word also means "sunlight". More importantly, one can see the possible development from Sami *päju* to Estonian *paike* with a possible suffix -ke at the end of the word.

Figure 6 shows the word query for "moon." As we saw in Section 4.2, in the Minoan language the moon is considered either the queen of stars or the head of stars. Therefore, we see the sequence 57-7-67, which means "star", appear in both definitions of "moon." In addition, the word "moon" appears also in some translated Linear A inscriptions. Finally, Figure 7 shows the word query for "star." It has some overlaps with the previous queries because of the above mentioned reasons.

5 DATA ANALYTICS

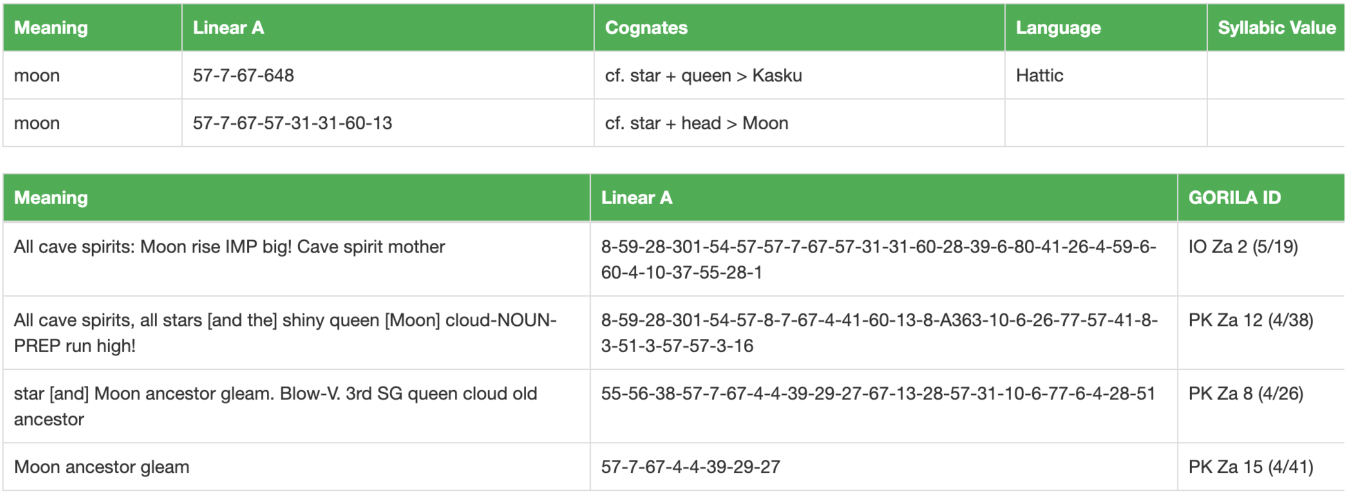
The AIDA system can do some simple data analytics. It can count the number of occurrences of any substring. It can also return the most frequent substrings of length k in the inscriptions database, where k is any integer greater than or equal to two. In the future we plan to extend these basic statistics to a more sophisticated analysis where the most frequent substrings are analyzed to check whether they occur preferentially in the beginning, the middle or the end of the inscriptions. This more sophisticated analysis could help determine whether the most frequent substrings are prefixes, word roots, or suffixes, and whether the root words are likely to be nouns



Meaning	Linear A	Cognates	Language	Syllabic Value
light	8-80	fény	Hungarian	fe-nu
light	8-80	bæggjo	Sami	fe-nu
light	8-80	päju	Sami	fe-nu
light	8-27	fény	Hungarian	fe-ne
light	8-27	bæggjo	Sami	fe-ne
light	8-27	päju	Sami	fe-ne
sunlight	302-344-28	paike	Estonian	pj-ai-ku
sunlight	302-344-28	fény	Hungarian	pj-ai-ku
sunlight	302-344-28	fehér	Hungarian	pj-ai-ku

Meaning	Linear A	GORILA ID
[Let the] cloud come, [the] Dan [river] flow, old Tamuz bring heat, shine sunlight	41-41-17-363-310-1-81-73-363-16-73-47-6-60-8-54-39-4-58-45-344-344-28	KN Zf 31 (4/155)

Figure 5: The result of querying the word "light".



Meaning	Linear A	Cognates	Language	Syllabic Value
moon	57-7-67-648	cf. star + queen > Kasku	Hattic	
moon	57-7-67-57-31-31-60-13	cf. star + head > Moon		

Meaning	Linear A	GORILA ID
All cave spirits: Moon rise IMP big! Cave spirit mother	8-59-28-301-54-57-57-7-67-57-31-31-60-28-39-6-80-41-26-4-59-6-60-4-10-37-55-28-1	IO Za 2 (5/19)
All cave spirits, all stars [and the] shiny queen [Moon] cloud-NOUN-PREP run high!	8-59-28-301-54-57-8-7-67-4-41-60-13-8-A363-10-6-26-77-57-41-8-3-51-3-57-57-3-16	PK Za 12 (4/38)
star [and] Moon ancestor gleam. Blow-V. 3rd SG queen cloud old ancestor	55-56-38-57-7-67-4-4-39-29-27-67-13-28-57-31-10-6-77-6-4-28-51	PK Za 8 (4/26)
Moon ancestor gleam	57-7-67-4-4-39-29-27	PK Za 15 (4/41)

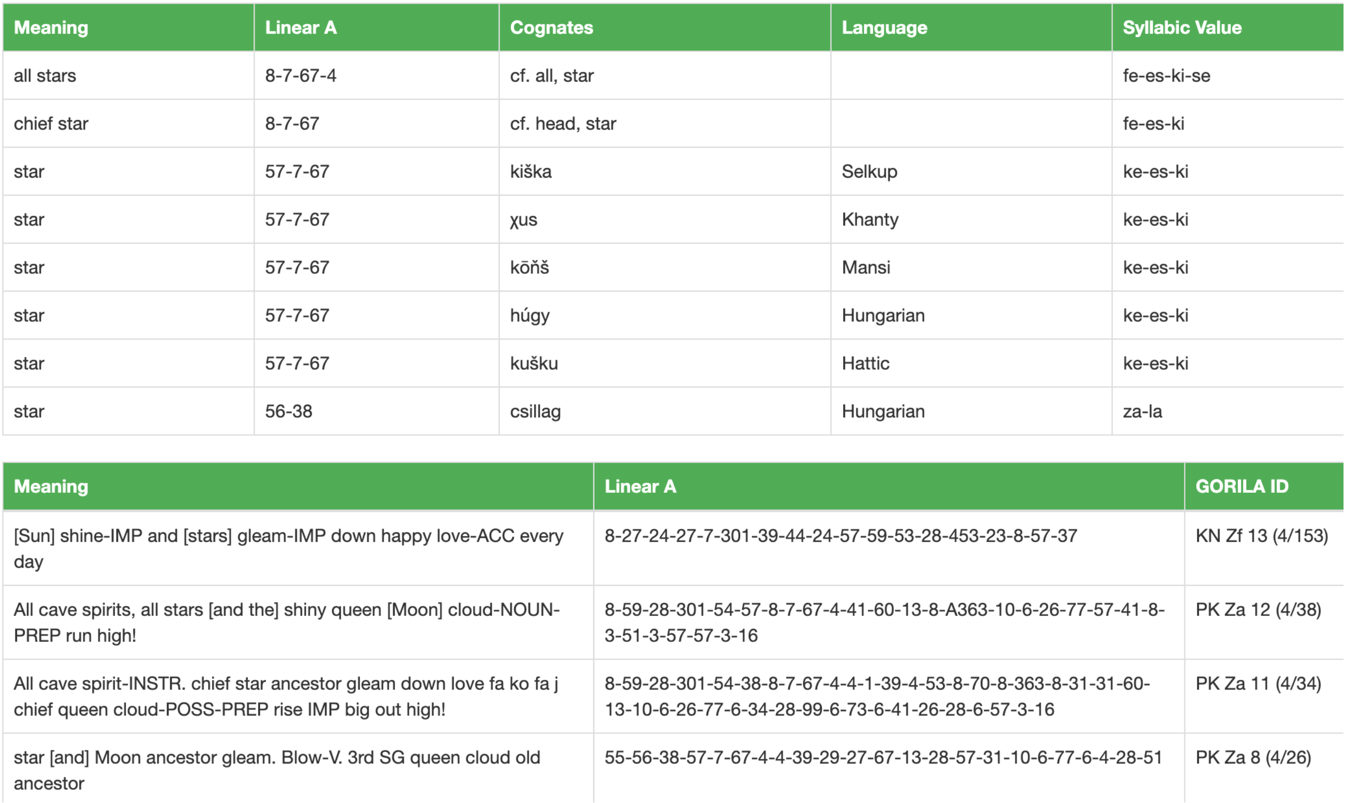
Figure 6: The result of querying the word "moon".

or verbs. The AIDA system also could help discover relationships among various scripts, strengthening recent work that shows that Near Eastern scripts have spread both to the west and to the east [4].

6 RELATED WORK

Currently, there is no other online Minoan inscription database system available for public use. However, there is a Linear B inscription database system called the DAMOS system, which is an abbreviation for *Database of Mycenaean at Oslo* [1]. The Linear B script was a successor of the Linear A script [11]. Linear B was the earliest form of Greek writing that is generally agreed to have been deciphered correctly in 1953 by M. Ventris and J. Chandwick [2, 19].

While not a database system, J. Younger's website at the University of Kansas, *http://www.people.ku.edu/ jyounger/LinearA/*, is a frequently consulted online resource for Linear A. It provides an online table of Linear A words with cross references, called "supports" on the website, to all the inscriptions in which the word occurs. Since this website is not a database system, it is not possible to look up in which inscriptions a word occurs by using a simple query. Instead a user needs to manually browse a list of Linear A inscriptions, which are provided on separate webpages, one for the



Meaning	Linear A	Cognates	Language	Syllabic Value
all stars	8-7-67-4	cf. all, star		fe-es-ki-se
chief star	8-7-67	cf. head, star		fe-es-ki
star	57-7-67	kiška	Selkup	ke-es-ki
star	57-7-67	χus	Khanty	ke-es-ki
star	57-7-67	kōňš	Mansi	ke-es-ki
star	57-7-67	húgy	Hungarian	ke-es-ki
star	57-7-67	kušku	Hattic	ke-es-ki
star	56-38	csillag	Hungarian	za-la

Meaning	Linear A	GORILA ID
[Sun] shine-IMP and [stars] gleam-IMP down happy love-ACC every day	8-27-24-27-7-301-39-44-24-57-59-53-28-453-23-8-57-37	KN Zf 13 (4/153)
All cave spirits, all stars [and the] shiny queen [Moon] cloud-NOUN-PREP run high!	8-59-28-301-54-57-8-7-67-4-41-60-13-8-A363-10-6-26-77-57-41-8-3-51-3-57-57-3-16	PK Za 12 (4/38)
All cave spirit-INSTR. chief star ancestor gleam down love fa ko fa j chief queen cloud-POSS-PREP rise IMP big out high!	8-59-28-301-54-38-8-7-67-4-4-1-39-4-53-8-70-8-363-8-31-31-60-13-10-6-26-77-7-6-34-28-99-6-73-6-41-26-28-6-57-3-16	PK Za 11 (4/34)
star [and] Moon ancestor gleam. Blow-V. 3rd SG queen cloud old ancestor	55-56-38-57-7-67-4-4-39-29-27-67-13-28-57-31-10-6-77-6-4-28-51	PK Za 8 (4/26)

Figure 7: The result of querying the word "star".

Haghia Triada inscriptions, another for the Knossos inscriptions, and so on at each separate location.

7 CONCLUSIONS AND FUTURE WORK

The development of the AIDA system is challenging because it requires knowledge of the important database system design principles as well as a knowledge of Minoan inscriptions and the basic concepts of comparative linguistics. These three areas of knowledge are uniquely brought together in our AIDA system. The AIDA system has a potential to be a widely used resource for many scholars in the humanities in the fields of classics, history and linguistics. As a future work, we hope to extend the system with other ancient languages, such as Sumerian [5, 18], Elamite [6], and the Indus Valley Script [3, 20]. As our database grows, we also investigate the possibility of using ElasticSearch [9] to make queries more efficient.

REFERENCES

[1] F. Aurora, A. Nesøen, D. Nedić, H. Løken, and A. Bersi. DAMOS - Database of Mycenaean at Oslo, 2018.

[2] J. Chadwick. *The Decipherment of Linear B.* Cambridge University Press, 1958.

[3] S. Daggumati and P. Z. Revesz. Data mining ancient script image data using convolutional neural networks. In *Proceedings of the 22nd International Database Engineering and Applications Symposium*, pages 267–272. ACM, 2018.

[4] S. Daggumati and P. Z. Revesz. Data mining ancient scripts to investigate their relationships and origins. In *Proceedings of the 23rd International Database Engineering and Applications Symposium*, 2019.

[5] C. Elisabeth and D. Caspers. Sumer, coastal Arabia and the Indus Valley in protoliterate and early dynastic eras: Supporting evidence for a cultural linkage. *Journal of the Economic and Social History of the Orient/Journal de l'histoire economique et sociale de l'Orient*, pages 121–135, 1979.

[6] R. K. Englund. The Proto-Elamite script. In P. T. Daniels and W. Bright, editors, *The World's Writing Systems*, pages 160–164. Oxford University Press, 1996.

[7] A. J. Evans. *Scripta Minoa: The Written Documents of Minoa Crete with Special Reference to the Archives of Knossos.* Classic Books, 1909.

[8] L. Godart and J.-P. Olivier. *Recueil des inscriptions en Linéaire A.* Number 21 in Études Crétoises. De Boccard, 1976.

[9] C. Gormley and Z. Tong. *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine.* O'Reilly Media, Inc., 2015.

[10] N. Marinatos. *Minoan Kingship and the Solar Goddess: A Near Eastern Koine.* University of Illinois Press, 2010.

[11] J.-P. Olivier. Cretan writing in the second millennium BC. *World Archaeology*, 17(3):377–389, 1986.

[12] J.-P. Olivier, L. Godart, and J.-C. Poursat. *Corpus Hieroglyphicarum Inscriptionum Cretae*, volume 31 of *Études Crétoises.* De Boccard, 1996.

[13] P. Z. Revesz. Bioinformatics evolutionary tree algorithms reveal the history of the Cretan Script Family. *International Journal of Applied Mathematics and Informatics*, 10:67–76, 2016.

[14] P. Z. Revesz. A computer-aided translation of the Cretan Hieroglyph script. *International Journal of Signal Processing*, 1:127–133, 2016.

[15] P. Z. Revesz. A computer-aided translation of the Phaistos Disk. *International Journal of Computers*, 10:94–100, 2016.

[16] P. Z. Revesz. Establishing the West-Ugric language family with Minoan, Hattic and Hungarian by a decipherment of Linear A. *WSEAS Transactions on Information Science and Applications*, 14:306–335, 2017.

[17] P. Z. Revesz. A translation of the Arkalochori Axe and the Malia Altar Stone. *WSEAS Transactions on Information Science and Applications*, 14(1):124–133, 2017.

[18] P. Z. Revesz. Sumerian contains Dravidian and Uralic substrates associated with the Emegir and Emesal dialects. *WSEAS Transactions on Information Science and Applications*, 16(1):8–30, 2019.

[19] M. Ventris and J. Chadwick. *Documents in Mycenaean Greek.* Cambridge University Press, 2nd edition, 1973.

[20] B. K. Wells and A. Fuls. *The Archaeology and Epigraphy of Indus Writing.* Archaeopress, 2015.

Indexing large updatable Datasets in Multi-Version Database Management Systems

Christian Riegger
Data Management Lab
Reutlingen University, Germany
christian.riegger@reutlingen-university.de

Tobias Vinçon
Data Management Lab
Reutlingen University, Germany
tobias.vincon@reutlingen-university.de

Ilia Petrov
Data Management Lab
Reutlingen University, Germany
ilia.petrov@reutlingen-university.de

ABSTRACT

Database Management Systems (*DBMS*) need to handle large updatable datasets in on-line transaction processing (OLTP) workloads. Most modern *DBMS* provide snapshots of data in multi-version concurrency control (*MVCC*) transaction management scheme. Each transaction operates on a snapshot of the database, which is calculated from a set of tuple versions. High parallelism and resource-efficient append-only data placement on secondary storage is enabled. *One major issue in indexing tuple versions on modern hardware technologies is the high write amplification for tree-indexes.*

Partitioned B-Trees (*PBT*) [5] is based on the structure of the ubiquitous B$^+$-Tree [8]. They achieve a near optimal write amplification and beneficial sequential writes on secondary storage. Yet they have not been implemented in a *MVCC* enabled *DBMS* to date.

In this paper we present the implementation of *PBT*s in *PostgreSQL* extended with *SIAS*. Compared to PostgreSQL's B$^+$–Trees *PBT*s have 50% better transaction throughput under TPC-C and a 30% improvement to standard *PostgreSQL* with *Heap-Only Tuples*.

CCS CONCEPTS

• **Information systems → Data access methods**.

KEYWORDS

Indexing Structure, MVCC, Modern Storage Hardware

ACM Reference Format:
Christian Riegger, Tobias Vinçon, and Ilia Petrov. 2019. Indexing large updatable Datasets in Multi-Version Database Management Systems. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3331076.3331118

1 INTRODUCTION

In times of Big Data, IoT, cloud computing and social media, datasets are large and and update-intensive. Database Management Systems (DBMS) are predestined to manage these datasets, but are the bottleneck of most data-intensive operations. Datasets have a near-linear growth and cannot be entirely located in main memory in most cases.

Whenever a transaction modifies a tuple in a multi-version concurrency control (MVCC) enabled DBMS such as PostgreSQL, a new version-record (tuple version or simply version) of this tuple is produced. Existing approaches in such DBMS organize versions as a doubly-linked list, where each version record has a *creation_timestamp* and an *invalidation_timestamp*, which is initially empty. Whenever a transaction TX_n creates successor version upon an update, both version records are modified, setting the *invalidation_timestamp* of the predecessor, and the *creation_timestamp* of the successor to *timestamp*(TX_n). We consider a novel version-organization, based on SIAS [3]. Every version has a *creation_timestamp*, and a single backward reference to its predecessor.

This version model assumes that every tuple comprises a set of tuple versions that are available as persistent version records, physically stored as a chain. While processing a query under a transaction, only the "visible" versions should be determined and passed on for processing. The Snapshot Isolation visibility criteria hold, i.e. a version $t_x.v_y$ is visible to transaction TX_4Q, if:

(1) $creation\_timestamp(t_x.v_y) = MAX(creation\_timestamp(t_x.v_{\{ALL\}})) < timestamp(TX_4Q)$;
(2) $transaction\_status(creation\_timestamp(t_x.v_y)) = COMMITTED$; and
(3) $creation\_timestamp(t_x.v_y) \notin L_{concurrent}(TX_4Q)$.

– Hence the version visibility check is very I/O intensive.

With this model, searching for one or a few data tuples with specific search predicates in base tables is an expensive operation with super-linear growth. In times of Big Data, when datasets typically cannot be entirely located in main memory, full table scans are not an option. *Indexes* describe an additional access path to tuples located in base tables. The index structure of a B$^+$-Tree [8] became ubiquitous in DBMS [7]. The tree-index allows accessing data in a key-sorted order in logarithmic time. Index record and structure maintenance operations in the sorted tree-structure cause a high write amplification (*WA*) to secondary storage – this effect is amplified by the maintenance of tuple versions. Characteristics of novel semiconductor-based storage technologies (fast reads, asymmetry, out-of place updates, high parallelism and wear) are not leveraged. *Indexing tuple versions is still an open research area, considering characteristics of modern storage hardware. Index structures need to handle modifications of index records out-of place for reduction of write amplification (WA) on secondary storage media.*

Partitioned B-Trees (*PBT*) [5] is based on the ubiquitous B$^+$-Tree [8]. They achieve a near optimal *WA* and beneficial sequential writes by collecting modifications in a main memory partition and forcing related nodes to secondary storage. Already persisted data is not physically affected by further modifications – i.e. maintenance of tuple versions in the index does not amplify *WA*.

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331118

In this paper we present *PBT*s in PostgreSQL extended with the version model of SIAS. Compared to PostgreSQL's B$^+$-Trees, *PBT*s achieve a 50% improved transaction throughput under TPC-C and a 30% improvement to standard *PostgreSQL* with *Heap-Only Tuples*.

The structure of this paper is as follows. We give an overview of related indexing approaches in Section 2. In Section 3 we discuss the conflict in design decisions of MVCC. We outline the algorithms of PBT in Section 4 and verify our assumptions in Section 5.

2 RELATED WORK

Most popular indexing approaches in DBMS are based on B$^+$-Trees, which can result in high write amplification (*WA*) on random updates in large datasets. PostgreSQL uses Heap-Only Tuples (HOT) as indirection layer to reduce index management operations. Index records reference items in base table, which point to tuple versions in the heap node. Corresponding tuple versions are located on the same node and are identified by processing the version chain. If a tuple version becomes garbage collected, the item is modified to reference the next relevant version. This indirection layer reduces index modifications, but cannot avoid *WA* of indexes. Furthermore the *WA* on base table nodes is increased for large datasets.

Maintaining out-of place tuple versions enable high parallelism and a beneficial append-only sequential write pattern to secondary storage for base tables. Snapshot Isolation Append Storage (SIAS) makes use of the natural append-only characteristics of tuple versions and achieves an increased throughput of 30% in comparison to PostgreSQLs standard base table organization in an OLTP workload [3]. Every version has a *creation_timestamp*, and a single backward reference to its predecessor. This version model assumes that every tuple comprises a set of tuple versions, which are available as persistent chain of version records. However, SIAS does not support HOT indirection layer, whereby indexing effort is increased.

2.1 MV-IDX

MV-IDX [4] is based on a B$^+$-Tree and maintains a virtual identifier for each tuple and in-memory data nodes for each version as an indirection layer. With Snapshot Isolation Append Storage (SIAS) [3], *WA* on base tables is reduced in comparison to HOT, but index management operations can cause in-place updates and a high *WA* – e.g. if an indexed attribute value becomes modified. Partitioned B-Trees (PBT) handle updates to indexed attribute values in a main memory partition in the PBT-Buffer, whereby *WA* is optimized.

2.2 Write Optimized B-Trees

Write Optimized B-Tree [6] aims to achieve an append-only write pattern in a B$^+$-Tree. It is organized like a traditional B$^+$-Tree with limited modifications – utilizing fence keys instead of sibling pointers enables this structure to perform out-of-place writes of modified nodes. Due to missing sibling pointers, cursors on scans cannot find siblings. Therefore, its sibling has to be requested in the parent node. Additional complexity in buffer management occurs.

Write Optimized B-Trees do not solve the problem of high *WA* in B$^+$-Trees. If a node gets evicted, it is written in a log-structure. However, a node is not protected from further modifications and already indexed data is written manifold. Partitioned B-Trees (PBT) collect modifications of leaf nodes in a PBT-Buffer until the partition gets evicted. Every record is written exactly once, except for garbage collection – *WA* is near optimal.

2.3 LSM-Trees

LSM-Trees [9] are optimized for high update rates and reduce *WA* due to collecting and pre-sorting modifications in a fixed-sized main memory component, which becomes evicted on a certain threshold and replaced by a new main memory component. As a result, several components exist on secondary storage media and are frequently merged in larger components. Pre-sorted records are migrated and sequentially written in a log-based pattern. bLSM-Trees [10] are based on the structure of LSM-Trees, however, there is a fixed count of three components for reduction of read amplification (*RA*). Furthermore, bloom filters protect components from unnecessary reads for point queries. Scheduling of merge areas and insertion rates between components reduce steals and replacement selection increases the effective amount of merged records.

Advantages of Partitioned B-Trees (PBT) are manifold. *First*, the single tree-index structure leverages the logarithmic relation between capacity and height of the tree. Index nodes are commonly used and buffered across partitions, whereby *RA* is reduced at same height like larger components in LSM-Trees. *Second*, compression methods, like suffix truncation, perform better in one large set of records, than in several smaller sets [2]. *Third*, partition sizes are self-balanced and workload adaptive due to commonly used PBT-Buffer. Managing component thresholds in LSM-Trees requires deep knowledge about the workload and administrative effort. *Last*, partitions of PBTs are more flexible than components in LSM-Trees. A partition can be created to absorb bulk loads with low effect on concurrent workload and merged or cropped from tree-structure afterwards, based on result of the transaction. Furthermore, *Cached Partitions* can be similar created out of result sets of frequently queried records and reduce *RA*.

3 MVCC TRANSACTION MANAGEMENT SCHEME

Multi-version concurrency control (MVCC) is the most popular transaction management scheme in modern DBMS. For instance, it is used by Oracle, MySQL-InnoDB, HyPer, SAP HANA, MongoDB-WiredTiger and PostgreSQL. In theory, MVCC enables high parallelism, because reading transactions do not block concurrently writing transactions. Modifications result in a new tuple version. Furthermore, in snapshot isolation, modifications of writing transactions do not block concurrently reading transactions, because for each transaction a visible tuple version can be returned.

The DBMS differently implement MVCC transaction management scheme. Fundamentals in design decisions are *(a) Concurrency Control Protocol*, *(b) Version Storage*, *(c) Version Ordering*, *(d) Garbage Collection* and *(e) Index Management* [11]. In fact, that *(a) Concurrency Control Protocols* deal with serialization strategies (first updater / committer wins) and has low effect on indexing and resulting write and read I/O patterns to secondary storage, we focus on and outline points *(b)* to *(e)* in the following. Afterwards, we give a short discussion. There is a conflict dilemma in usage of the optimal design decisions for large datasets and characteristics of modern storage technologies.

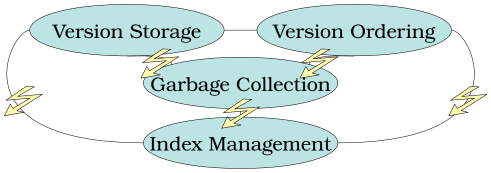


Figure 1: Dilemma: Conflicts in Design Decisions

3.1 Version Storage

Tuple versions correspond to one logical tuple. They form a linked list, which represents a version chain. Mainly, a version of a tuple can be maintained in two different ways – logically or physically. The first type means that for each modification of a logical tuple a delta record indicates the difference to another version. These delta records are connected and required to restore a tuple version. Physical storage means, that each tuple version is entirely stored.

In both cases, following information is required: a *content / delta* that is stored in the DBMS for a tuple, logical *timestamps* for *validation* and *invalidation*, and a *reference* to its *predecessor* and / or *successor version* – based on the version ordering. Modifications are performed in-place or out-of-place.

Considering the characteristics of modern storage technologies, out-of-place updates are preferable due to less write amplification (*WA*) to secondary storage for large datasets. Furthermore, this behavior enables higher parallelism than in-place updates, whereby the tuple version has to be exclusively latched for modification. This is possible with logical and physical version storage. Delta records tend to consume less space than physical tuple versions, but require further versions for tuple reconstruction.

3.2 Version Ordering

A version chain of a logical tuple forms a linked list. A doubly linked list enables knowledge of predecessor and successor, however, it requires additional latches and reduce parallelism on modification [11] in comparison to singly linked lists, which require a version ordering. Discovering the visible tuple version to a transaction snapshot requires to follow the version chain from an entry point, until the visible version was found. Basically, there are two different ordering methods – old-to-new (O2N) and new-to-old (N2O) for singly linked lists. For both methods in-place as well as out-of-place updates are possible. In case of O2N-ordering, the entry point is the oldest tuple version in version chain. A visibility check requires to process all predecessors, beginning from the oldest tuple version. Updates (insertion of tuple versions) require at least modifications of predecessors *invalidation_timestamp* and *reference*. N2O-ordering means that the entry point is the most recent tuple version, which references to its predecessor. Queries in OLTP transactions can find the visible version very well, because the most recent tuple version is the entry point of the version chain.

Considering the characteristics of modern storage technologies N2O-ordering for physical version storage result in best *WA* with append-only characteristic for large datasets, because maintenance of *validation_timestamps* of recent versions are sufficient. Other combinations require in-place updates, which shrink benefits in parallelism of a singly-linked list and *WA*.

3.3 Garbage Collection

Tuples are modified multiple times. In MVCC modifications result in successor versions. Predecessors become obsolete, if they are no more visible for any active transaction. Garbage collection (GC) reclaims space and can improve *RA*, especially for O2N version ordering. However, GC increases *WA* on secondary storage. Tuple level GC can be performed as background vacuum and cooperative cleaning process [11]. In the first case, a background thread scans and purges obsolete versions. Cooperative cleaning uses the process operation of version chains for detection of obsolete tuple versions. GC operations have to minimize effects on *WA* and additional access paths – e.g. if the entry point changes, indexes require adaptions.

3.4 Index Management

Complexity of index management strongly depends on version storage and ordering techniques. A lossy result from index scans is not acceptable, so in theory every tuple version should be indexed. This approach can result in massive *WA* and *RA*, e.g. in case of B^+-Trees. Most popular indexes do not support visibility checks in MVCC. Therefore, the version chain of the base table is required to determine the visible tuple version. As a result, at least the entry point of the version chain is indexed. Modifications in the tuple versions content, which affect search key columns have to become visible to an un-lossy index. There are two possibilities to map index records to tuple versions in base tables. First, physical references – the entry point tuple version in base tables can be directly accessed, but changes to the entry point location result in index modifications. Second, an indirection layer with logical references is implemented. Therefore, each version of a tuple is referenced with an unique identifier. Index records reference to this unique identifier in the indirection layer, which references the entry point version location. This approach can reduce index modifications.

In-place updates of tuple versions in base tables reduce index maintenance operations with physical indirection. However, as outlined in Sections 3.1 and 3.2, an out-of-place append-only scheme brings benefits on modern storage technologies. Indexing tuple versions with an indirection layer can reduce index maintenance of the preferred storage scheme, if the search key attributes of the tuple content remain constant. However, inserting and modifying tuples in a traditional strict alphanumeric-sorted index structure result in in-place updates on index nodes and high *WA*.

3.5 Discussion

We outlined relevant design decisions for storing tuple versions in MVCC transaction management scheme. Every design brings benefits for specific tasks and requirements. We focus on large update-intensive datasets, which cannot be entirely located in main memory. Therefore, we introduced the dilemma in different designs.

Modifications are preferably stored as physical tuple versions in base tables, due to tuple reconstruction costs. Out-of-place updates reduce *WA*. This can be achieved by a new-to-old (N2O) version ordering, because *invalidation_timestamps* of predecessors can be reconstructed from previously processed successors *creation_timestamps* and predecessors remain constant. Garbage collection (GC) is required for space reclamation, but brings additional complexity to data structures. Effects on indexes and *WA* should be minimized.

A N2O version ordering requires index maintenance for every new tuple version, because the entry point of the version chain changes. A logical indirection layer could reduce index maintenance effort, however, a sequential write pattern to secondary storage cannot be achieved with traditional indexing structures. *Due to high update rates to indexes, caused by insertion of new tuple versions on modifications, traditional index structures become a bottleneck.*

We decided to implement Partitioned B-Trees (PBT) in PostgreSQL extended with SIAS[3], due to its beneficial append-only write I/O properties to secondary storage. We describe the structure and algorithms of PBT and how it is able to achieve the preferable sequential write pattern to secondary storage.

4 APPROACH: PARTITIONED B-TREES

Partitioned B-Trees (PBT) [5] are based on traditional B^+-Trees [8] and make use of its intrinsic and well studied algorithms with few modifications. The essential difference is an introduced artificial leading key column – the partition number. An index record consists of a *partition number*, its *search key columns* and a *physical tuple reference* or an *unique virtual identifier* for tuple assignment. Every different partition number value describes a single partition. This enables the PBT to maintain partitions within one single tree-structure in alphanumeric sort order. Partitions can support additional functionalities, like reorganizations or bulk loads [5].

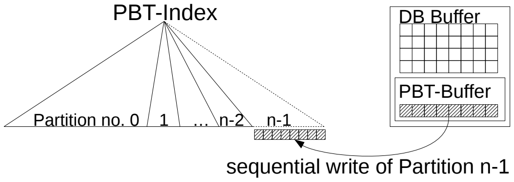


Figure 2: Sequential write of a Partition

PBTs write any modification of index records exactly once on eviction of a partition, except for later reorganization or garbage collection operations, what enables a beneficial sequential write pattern to secondary storage. This is realized by evicting all related leaf nodes of a partition. Leaf nodes of modifiable main memory partitions are stored in a separate area of the buffer cache – the PBT-Buffer. Records can only be inserted or updated in partitions, which are located in the PBT-Buffer. In case of full PBT-Buffer, a main memory partition is written to secondary storage: *First*, a new partition is created to support ongoing modifications and the partition, which has to be evicted, becomes immutable. *Second*, a bloom filter and prefix bloom filter is created, gets filled with all index records in the recently closed partition. *Last*, all leaf nodes are sequentially written to secondary storage.PBT indexes in MVCC are not lossy, however, they are able to return a set of entry points to candidate tuples, which have to be verified in a visibility check. We describe the index operations in a PBT:

Insert Operations. are only performed in a mutable main memory partition. Therefore, the first search key column is prepended with its partition number. The index structure is traversed and the index record is inserted at its regular position in the B^+-Tree structure. The leaf node is guaranteed to be located in main memory. Uniqueness constraints are supported by first performing a read operation.

Algorithm 1 Partitioned B-Tree - INSERT

```
Input: Regular $|attr_{val}|$, $ref$
Output: $ErrCode$
procedure INSERT($|attr_{val}|$, $ref$)
    Let $part_{insert} \leftarrow$ MAX($PartitionsList$)
    Let $part\_rec \leftarrow$ FORM_PART_REC($part_{insert}$, $|attr_{val}|$, $ref$)
    UNIQUENESS_CONSTRAINT_CHECK($|attr_{val}|$)        ▷ check all $Partitions$
    return DO_REGULAR_INSERT($part\_rec$)
```

Update Operations. can be performed in-place, if the index record is still in a mutable main memory partition. Therefore, only the physical tuple reference field has to be modified. If the index record of the updated tuple is in an evicted immutable partition on secondary storage or search key columns are affected, an insert in a mutable main memory partition is performed.

Algorithm 2 Partitioned B-Tree - UPDATE

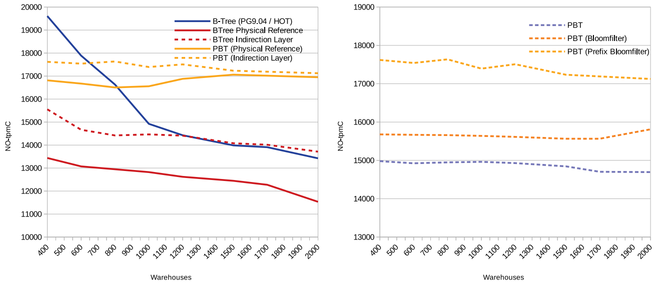
```
Input: Regular $|attr_{val,old}|$, $|attr_{val,new}|$, $ref$
Output: $ErrCode$
procedure UPDATE($|attr_{val,old}|$, $|attr_{val,new}|$, $ref$)
    Let $part_{update} \leftarrow$ MAX($PartitionsList$)
    Let $part\_rec \leftarrow$FORM_PART_REC($part_{update}$, $|attr_{val,old}|$, $reference$)
    if $|attr_{val,old}| = |attr_{val,new}|$ and FIND($part\_rec$) then
        return $in\_place\_update$($|attr_{val,old}|$, $ref$)
    else
        return INSERT($|attr_{val,new}|$, $ref$)
```

Delete Operations. are performed similar to update operations. The physical tuple reference points to a tombstone record.

Algorithm 3 Partitioned B-Tree - SCAN

```
Input: Regular $|attr_{val,min}|$, $|attr_{val,max}|$
Output: $|refs|$
procedure SCAN($|attr_{val,min}|$, $|attr_{val,max}|$)
    for each $part_{scan} \in PartitionsList$        ▷ start MAX($PartitionList$)
    Let $part\_rec_{min} \leftarrow$ FORM_PART_REC($part_{scan}$, $|attr_{val,min}|$)
    Let $part\_rec_{max} \leftarrow$ FORM_PART_REC($part_{scan}$, $|attr_{val,max}|$)
    if $|attr_{val,min}|..|attr_{val,max}| \in part_{scan}.filter$ then
        Let $ref$ ←FIND_IN($part\_rec_{min}$, $part\_rec_{max}$)
        $|refs|$.ADD($ref$)
        loop
            if not HASNEXT( ) then
                break
            Let $ref$ ←NEXT( )
            $|refs|$.ADD($ref$)
    return $|refs|$
```

Search and Scan Operations. are not allowed to be lossy, however, they return a set of candidate tuples, which have to be verified in a visibility check in base table. The query search predicates are modified to match the search key columns in a PBT – a partition number is prepended to the first search key column. The partitions in a PBT are traversed and scanned from the highest to the lowest numbered partition. This behavior is beneficial for performing reads on unique search key column values. If a matching index record was found, further lower numbered partitions do not have to be processed and the algorithm can break up earlier whereby *RA* is reduced. The returned candidate tuples are send to the visibility check in base table. Order requirements for scans are processed afterwards. Read and scan operations can be accelerated by filter techniques. (Prefix) bloom filters reduce *RA* and latencies of point and range queries.



(a) Logical vs. Physical Reference

(b) Influence of Filter Techniques

Figure 3: OLTP Benchmark Throughput Evaluation

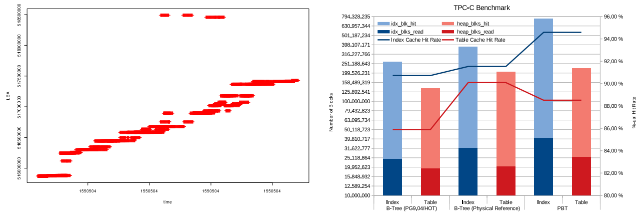
5 EXPERIMENTAL EVALUATION

We show Partitioned B-Trees (PBT) in comparison to traditional B$^+$-Trees in PostgreSQL; a RDBMS with MVCC transaction management scheme. PostgreSQL uses an O2N version order and physical tuple version storage. Index records have a physical reference to items located in base tables – denoted as B-Tree (PG9.04/HOT). PostgreSQL base table storage was modified to SIAS with a beneficial append-only write pattern and N2O version ordering. We evaluated B$^+$-Trees and PBT with physical and logical tuple reference.

We deployed the DBMS on an *Ubuntu 16.04* server with *Intel(R) Xeon(R) 3.50GHz* processor and an *Intel SSD* secondary storage device. We used the well-known DBT-2 [1] OLTP benchmark.

First, we evaluate throughput of B-$^+$Tree (PG9.04/HOT) as well as SIAS with B$^+$-Trees with physical and logical reference in the DBT-2 benchmark. In Figure 3a, we show the throughput for different dataset sizes. The buffer cache of the DBMS is set to 600MB. The dataset size increase with the warehouse count. B-$^+$Tree (PG9.04/HOT) performs well, if most buffers are located in main memory. Updates are performed in base tables by HOT. The index maintenance effort is low, due to this indirection. If the workload becomes write-intensive, the throughput falls rapidly. SIAS has a scalable throughput [3], but increased effort in index management shrinks performance with physical reference B$^+$-Tree updates. With an indirection layer, index management is reduced to inserts and updates of search key columns, whereby the throughput is increased by up to 20% and SIAS performs better than PG9.04/HOT at 1200 warehouses. Effects of indirection layer on index management are minimal for PBT. The throughput difference is 6% at the dataset of 1000 warehouses. As the dataset grows, there is almost no difference in throughput between PBT with physical and logical reference. The index is able to absorb additional modifications. PBT with SIAS has a 50% increased throughput in relation to comparable B$_+$-Trees with physical references and about 30% with indirection layer at 2000 warehouses. The append-only approaches (SIAS and PBT) outperform PG9.04/HOT, as the benchmark becomes write-intensive at 700 warehouses – up to an improvement of 30% at 2k warehouses.

Partitioned B-Trees (PBT) append modifications to the dataset in a main memory partition. Effort of look-ups and especially of scans increase by number of partitions (see Figure 3b), because in theory every partition has to be traversed. Up to 25 partitions were created for update-intensive indexes over the test duration. Point queries can break look-up on first matching record, which is visible to a transaction snapshot. Point queries can skip partitions, based on bloom filters and increase throughput up to 10%. The benchmark includes several scans. Prefix bloom filters include a fixed set of scan attributes and increase total throughput by another 10%.



(a) Sequential Write Pattern of PBT

(b) Requests / Cache Hit Rate

Figure 4: Evaluation of Index Properties

We evaluated the write pattern of PBT (see Figure 4a). The diagram indicates the eviction of a main memory partition to secondary storage. Each red cross indicates the write of an index node. Several parallel and sequential writes of extends are shown in the diagram. Once an index node was evicted to secondary storage, its contents never change. PBT achieves the desired beneficial write pattern.

In Figure 4b we show the requests on index nodes (blue) and base table nodes (red) for an write-heavy OLTP benchmark. Requests on cached nodes are displayed brighter than fetches from secondary storage. The results are calculated for equal throughput over the test duration and all tables and indexes. PBT requires more requests on index nodes due to partitioning of index records and larger record sizes. Most requests are on buffered nodes, because many queries can be answered in the main memory partition. Index records of recent tuple versions are common to be located there. Requests can benefit from better cache hit rate in comparison to B$^+$-Trees.

6 CONCLUSION

We presented different design decisions in MVCC transaction management scheme for large-scale data sets and update-intensive OLTP workloads, regarding the characteristics of modern storage technologies. We outlined resource-efficient append-only version-organization in base tables and its conflict with index management. We firstly implemented Partitioned B-Trees in a DBMS with MVCC transaction management scheme and evaluated their throughput and characteristics. PBT achieves an up to 50% increased throughput in relation to comparable B$^+$–Trees.

ACKNOWLEDGMENTS

This research was partially funded by the Ministry of Science of Baden-Württemberg, Germany, for the Doctoral Program 'Services Computing'.

REFERENCES

[1] Database Test Suite - Browse /dbt2 at SourceForge.net. Accessed: 2019-03-01.
[2] R. Bayer and K. Unterauer. Prefix b-trees. *ACM Trans. Database Syst.*, 1977.
[3] R. Gottstein. *Impact of new storage technologies on an OLTP DBMS, its architecture and algorithms.* PhD thesis, Technische Universität, Darmstadt, 2016.
[4] R. Gottstein et al. Mv-idx: Indexing in multi-version databases. IDEAS '14, 2014.
[5] G. Graefe. Sorting and indexing with partitioned b-trees. In *CIDR*, 2003.
[6] G. Graefe. Write-optimized b-trees. pages 672–683. VLDB Endowment, 2004.
[7] G. Graefe. Modern b-tree techniques. *Foundations and Trends in Databases*, 2011.
[8] P. Lehman et al. Efficient locking for concurrent operations on b-trees. 1981.
[9] P. O'Neil et al. The log-structured merge-tree (lsm-tree). 1996.
[10] R. Sears et al. blsm: A general purpose log structured merge tree. SIGMOD, 2012.
[11] Y. Wu et al. An empirical evaluation of in-memory multi-version concurrency control. *Proc. VLDB Endow.*, 2017.

An *Neo4j* Implementation for Designing Fuzzy Graph Databases

Bruno Ponsoni Costa
Computer Science MSc Program
UNIFACCAMP
Campo Limpo Paulista
Brazil
bruno.ponsoni@ifsp.edu.br

Luis del Val Cura
Computer Science MSc Program
UNIFACCAMP
Campo Limpo Paulista
Brazil
delval@faccamp.br

ABSTRACT

Imperfect data express their meanings incompletely and the Theory of Fuzzy Sets arises as mathematically support for the interpretation of those data. The union of these concepts describes a new data type, called fuzzy data. We discuss the use of fuzzy data in Graph Databases. Previous works define fuzzy queries on Graph Databases but the data stored is a regular and perfect data. In that works we extent a Graph Database and lets the users store information in the fuzzy and imperfect data. The databases management system *Neo4j* is proposed to developed the application and the *Cypher* database languages to describes the imperfect data definitions. We uses a social network use case to illustrated the works,

CCS CONCEPTS

• **Information systems → Database design and models**; Graph-based database models.

KEYWORDS

Fuzzy Graph Database, NoSQL Database

ACM Reference Format:
Bruno Ponsoni Costa and Luis del Val Cura. 2019. An *Neo4j* Implementation for Designing Fuzzy Graph Databases. In *23rd International Database Engineering Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3331076.3331082

1 INTRODUCTION

More and more applications are now being made that use data captured in real time, from different sources and formats. These data do not always have the clarity and complete meaning, admitting subjective interpretations, or, still, needing complements, so that their comprehension is possible. Due to these characteristics, these were termed as *imperfect data* [5]. The theory of fuzzy sets arises as a way to aid in the interpretation of these data, since it modifies the traditional Boolean concept [15] [7]. Traditionally an element may or may not belong to a data set, represented mathematically by *0* or *1*. In the theory of fuzzy sets this concept is expanded, allowing to define how much an element can belong to a set. A value between *0* and *1*, is assigned to the element, in order to represent the relevance this element has in front of the related set. In this way, it became possible to mathematically associate an imperfect data with a set of possible interpretations. The data generated by the union of these concepts are referred to as *fuzzy data*.

The patterns of traditional databases structures do not support, in their totality, storage and manipulation of fuzzy data [9]. This is due to the complexity that exists in relating a fuzzy data to an crisp data. Various approaches have been developed in order to offer adequate support to this type of data, in the most varied database structures. However, the methods developed completely meet the requirements, or even become complex depending on the type of the data and the structure that will carry it.

Graph databases [1] have gained prominence over the various existing data structures. Largely, because of its flexible feature, which makes it possible to represent the complex relationship between different types of data. It also has a management system, the *Neo4j*, that facilitates the visualization of the data structure and its relationships. This system is considered the favorite among developers, it offers a quick and efficient support in the handling of data, with its own easy-to-understand language, the *Cypher*. Given this, the possibility of incorporating the fuzzy data in this model is questioned. Since unstructured data models allow the interaction between different types of data.

The purpose of this article is to present a method that incorporates fuzzy data into a database, allowing the insertion of different types of data. Due to the complexity of relating different types of data in traditional database models, we consider the use of graph databases here. The advantages of the graph model, as compared to the manipulation of complex data, when compared to other models of databases, caused in the choice of this one. Besides being a model little explored in the literature. Thus, this article present an application developed for the use of fuzzy data in graph databases, both in data structure and in the query and insertion instructions. It seeks to complement the *Neo4j* system, allowing imperfect data to be used, based on the theory of fuzzy sets. The remainder of this article is organized as follows. Section 2 presents the definition of concepts belonging to fuzzy sets and imperfect information. Some important aspects of the graph databases as well as the *Neo4j* management system are presented in section 3. Section 4 we have the presentation of the problem in more detail and the way the proposed application is applied as a solution of these. Section 5 presents the results obtained in the application of a use case, as well as expectations for the further improvement of the application.

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331082

2 IMPERFECT INFORMATION AND FUZZY SETS

Imperfect information is related to incomplete, inaccurate or vague data. Understanding of stored data can be some complex, because the difficult to establish the relation between data of different types, even if they make reference to a same set of information. Imperfections in information are classified into five main types: *imprecision*, *uncertainty*, *vagueness*, *ambiguity*, and *inconsistency* [6]. The occurrence of one type of imperfection does not excludes the possibility of a second or third type, related to the first. As an example, *"I'm 95% certain that John was born in the 90's"*. It demonstrated the occurrence of *uncertain* along with *imprecision*, respectively, in the same information.

Imperfect data should not be interpreted solely as true or false boolean values. The definition or interpretation of imperfect data is conditioned to the subjectivity of the user, based on aspects experienced by the user. Factors such as place of birth and age are examples of aspects that affect the interpretation of data. For example, the classification of a person of *52 years* of age is relative, since, from the point of view of a child, most adults are classified as elderly. A second case consists of a person's height rating. In a certain region where it is common for the average height of the population to reach at most *170cm*, a person with this height is considered *high.* However, in another region with a higher average height, this same height can be considered only as *median.*

The theory of fuzzy sets assists the definition of imperfect data. Rather than forcing the assignment of an exact value to the imperfect data, we must mathematically classify its possible interpretations for their relevance. Fuzzy sets, initially proposed by [15], is related to the concepts of imperfect information. This expanded traditional Boolean concepts, which considers that an element can only belong or not to the set. In the theory of fuzzy sets, a coefficient of membership to an element can be assigned, thus representing how much that element belongs to the set. Thus, where U is a set of u elements of discrete or continuous universe. A ***fuzzy set*** F in U is characterized by a ***membership function*** represented by $\mu_F(u)$, which associates each element of U with values âĂźin the range $[0, 1]$. The set F can be expressed by the set of ordered pairs of U, that is, $F = \{(u, \mu_F(u)) | u \in U\}$. The ***support set*** of F is defined as a subset of U with degree of membership greater than 0, such that: $Supp(F) = \{u | u \in U, \mu(u) > 0\}$. An ***inflection point*** on F is the element whose membership value is: $\mu_F(u) = 0,5$, considered the greatest point of uncertainty of the set. The $\alpha - cut$ on F is a threshold level, considered valid elements, provided that they are equal or above that value. This must have the degree of membership above 0 to 1, that is: $F\alpha = \{u | \mu_F(u) \geq \alpha\}$ to $0 \leq \alpha \leq 1$. Elements defined by a $\alpha - cut$ with value close to 1, are the elements considered the most belonging to the set. In general there are several functions that can be used to obtain a membership degree. Among the most common are the *trapezoidal* and *triangular functions* [15]. The *trapezoidal* function is expressed for the quadruple (A, B, C, D), where $C(F) = [B, C]$ *and* $S(F) = [B - A, C + D]$.

The *trapezoidal function* classified a *52-year-old* is an *adult* or and *elderly* person. The proposed age has 0.2 of membership degree with the term *elderly* and 0.8 with the *adult* term, making the classification more compatible.

The fuzzy logic set with the concepts of imperfect data has given rise to the so-called fuzzy data. Fuzzy data defined from the fuzzy logic, its interpretations can be determined mathematically.

3 GRAPH DATABASES

The representation of information through graphs has been used by several models. A graph G is composed of vertices V and edges E, formally expressed by: $G = (V, E)$. Basic concepts about the various models of graph databases, their aspects and comparisons, can be seen in [13], [4], [8], [1] e [2]. The properties graphs has been adopted by most developers, as standard model for graph databases. In that graphs, the attributes are basically defined to vertex and edges properties and we can define the more complete representation of the relation of the data. Graph databases has been used for semi-structured models, such as *XML*, [2] and the interest has grown, since it has the capacity to support complex data structures.

The management system *Neo4j*[2] is considered most popular by developers of graph databases, [4]. This is due to the support offered to different programming languages, besides perfectly integrating the concepts belonging to the databases of property graphs. The *Neo4j* further includes a proprietary manipulation language, the *Cypher.* The main philosophy of the *Cypher*is to have a simple and intuitive syntax language, clear feature for easy reading and understanding, [12]. This has increased the interest in language, because unlike the other traditional models, the complexity in the development is not related to the intimacy in the use of these languages.

3.1 Fuzzy Graph Databases

A traditional graph database is composed of exact data represented by vertices or properties of these and the relationships that exist between these data, represented by edges. Integration with fuzzy logic caused the emergence of databases fuzzy graphs. This model may represent especially data where relationships are not fully defined. An overview of concepts about fuzzy graphs is presented in [14].

In fuzzy graphs the fuzzy information can occur both in the data existing in the attributes of the vertices and in the structure of the relationships of a graph database, [11] e [10]. In this way, it indicates that there is a need to allow fuzzy data to be entered as property values of a vertex, as well as that a fuzzy data is used to define a relationship. This insertion causes the Boolean view of whether or not a particular relationship exists to be reinterpreted. We can then analyze a path or distance between vertices in a differentiated way, being influenced by the definition of these fuzzy data.

In Figure 1 of [11], there is the occurrence of fuzzy data. It is possible to notice that the relationships of the vertices of the graph have values that modify their understanding. The relationship of the node *"Pierre"* with the vertex *"Serge"* is defined by an edge labeled *"contributor"* of value *"0.04"*. Considering the other edges with *"contributor"* labels and their values, we have identified that the highest value assigned is the value *"1"*. Any value below the value *"1"* is then considered to be a relationship of lesser intensity.

[2] *https://neo4j.com*

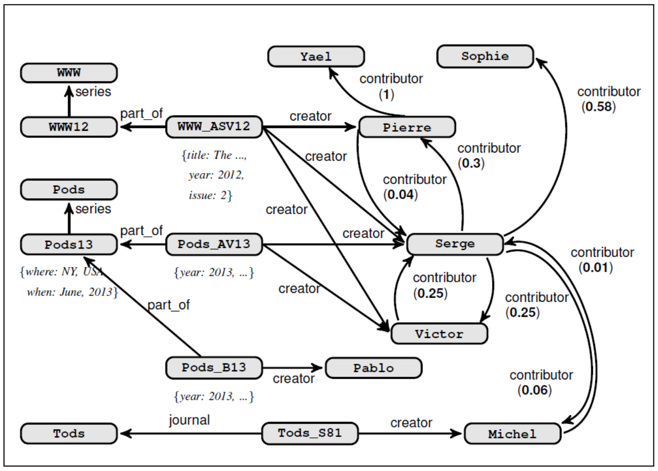


Figure 1: Representation Model of the fuzzy graph.

It must also consider in the model of fuzzy graphs its capacity in support both perfect and imperfect data types. The model by [11] has these concepts, offering equivalent support both for imperfect as perfect information. The Figure 1, demonstrates these concepts through their edges.

Due to the nature of the graph database model, the representation of the complex information is simplified. The greatest complexity is that can relate a fuzzy data to a precise or at least interpretation. Therefore, a large part of the studies carried out with the objective of to incorporate the fuzzy logic in graph databases are directed to the execution of instructions. These queries should assist users in obtaining the regardless of the form in which this information is stored.

4 A FUZZY DATA DEFINITION

Based on the proposals of [10] and [3], this work presents an application developed in order to work on the integration of fuzzy data. Unlike the above-mentioned works, the application will not be limited to the execution of nebulous queries, which seek to obtain data from a flexible syntax.

In that model, we aim to incorporate new functionalities, such as the insertion of fuzzy data and the execution of both flexible and traditional queries, performed in perfect or fuzzy data. Was used *RabbitHole*[1] based on the *Neo4j* system. The reason for using this system is due to the need to modify the system internally to accept the new types of data. As programming language was used the language *Java*[3] in conjunction with the *Apache Maven*[3] library.

The proposed environment allows the modification and testing of new functionality, without compromise the functions of the official system. In this way, we develop an algorithm for data interpretation, for query and insert instructions containing fuzzy data. It is necessary to *"teach"* the system about some fundamental parameters, to analyze a fuzzy data. Was developed a second algorithm, responsible for storing the definitions declared by the users about the fuzzy data. That document, provides for the interpretation algorithm to be executed. Finally, it was necessary an algorithm to measure and classify the relation of the data declared with the obtained data. The classification algorithm evaluates the membership degree measured in the comparison of the data declared in the instructions with the data obtained in response. The classification is only used with instructions of queries and not of insertion.

In general, to enable the insertion or queries fuzzy data, or even the classification of the answers obtained, we must to inform the system of the parameters necessary for the user to understand them. Unlike the [10] model, defining fuzzy data at the time of execution of instructions would not be feasible, since it would be necessary to define each possible given type existing in the model. Thus, it must be store the fuzzy data definitions, so that they could be used later. The structure generated for in that case is a *Fuzzy Data Definition Document* (*FDDD*). That structure, separated from the system, allows modified algorithms to query the fuzzy data defined by the users. In addition, it enables the sharing of settings for use by other systems or users. In this proposal, the *XML* document was used as the storage base for the definitions of the fuzzy data, because it has the ability to be understood by several systems, in addition to the library *Dom4j*[5] *Java* which has several manipulation resources of this type of document. In this way, integration with the system was benefit.

For the description of the fuzzy data, we extended the *Cypher* language to understand the developed of fuzzy data. In this way, a user can define this data directly in the application and at run time. The use of *Regular Expressions* helps in reading the syntax, making this new functionality possible. An example of the fuzzy data definition statement syntax is presented in the following statement.

DEFINE NODE_PROPERTY linguistic_variable fh *age AS young* =
{10, 20, 30, 40}

Each type of imperfection has a different form of interpretation. Therefore, it is not possible, so far, to define a generic type of imperfect data. Thus, an interpretation syntax has been developed for each data type. In Figure 2, we have the relation of the data types and the syntax definition, as it interpreted by the system. That relation is storage in *FDDD*.

The proposed classifications allow the application to identify which group of possible interpretations the data belongs to. This allows a relation between this data and others in the same group to be established, considering that they have a certain affinity.

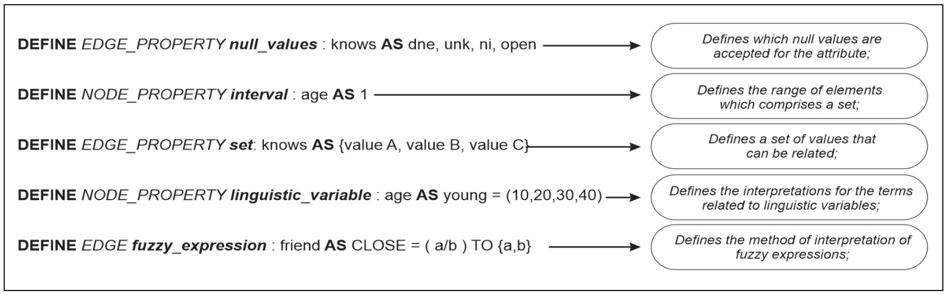
Each group uses a specific type of function, such as trapezoidal, triangular, among others, developed to obtain the degree of pertinence of the relation of data associated with it.

The interpretation will be based on the classification of the type to which the cloudy data is associated, its position in the graph and the base parameters informed in the definition of the data type.

In this way, it allows the application to interpret and associate different types of data, perfect, imperfect or imperfect that have different types of imperfection.

The proposed classification model has the purpose of allowing different types of imperfect data to be used, since these can be defined by their own users.

[1] *https://github.com/neo4j-contrib/rabbithole*
[3] *https://www.java.com*
[3] *https://maven.apache.org*
[5] *https://dom4j.github.io/*



DEFINE *EDGE_PROPERTY* ***null_values*** : knows **AS** dne, unk, ni, open

DEFINE *NODE_PROPERTY* ***interval*** : age **AS** 1

DEFINE *EDGE_PROPERTY* ***set***: knows **AS** {value A, value B, value C}

DEFINE *NODE_PROPERTY* ***linguistic_variable*** : age **AS** young = (10,20,30,40)

DEFINE *EDGE* ***fuzzy_expression*** : friend **AS** CLOSE = (a/b) TO {a,b}

Defines which null values are accepted for the attribute;

Defines the range of elements which comprises a set;

Defines a set of values that can be related;

Defines the interpretations for the terms related to linguistic variables;

Defines the method of interpretation of fuzzy expressions;

Figure 2: Defining syntax for imperfect data types.

This causes the limit of acceptance of which data can be used in a database system to be expanded because the limit will be determined only by how much the application can interpret from this data.

The classifications that refer to the group of the imperfect data type were designed with the purpose of helping the application to read and interpret the data used.

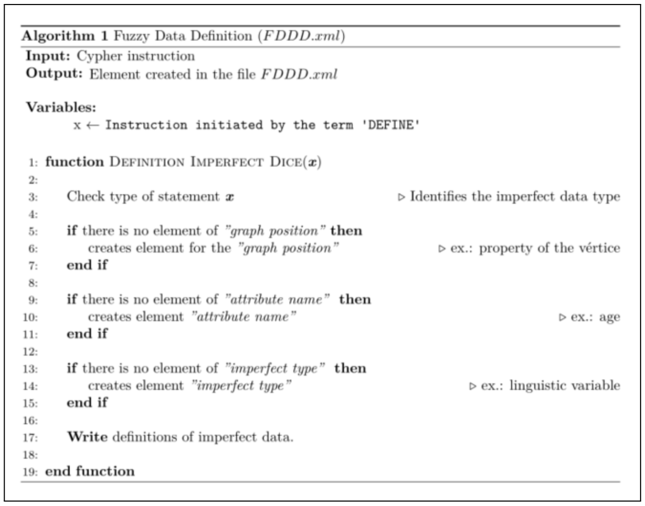
This classification is essential to avoid that the application obtains duplication of interpretations. Because, an imperfect data may be similar to another, but may have different interpretations, depending on the context in which it is used.

In Figure 3 is presented the algorithm to define the different fuzzy data types with an step-by-step to define the imperfect data. It should be checked if the element definition already exists, because this avoids duplication of the definitions, in that case, is prevent different interpretations for the same imperfect data type. Otherwise, it could not be determined exactly what the real interpretation is for this. The application would always use the first definition, ignoring the others.

Finally, a data structure with the definitions declared by the users will be generated, according to Figure 4. They are grouped the positions of the graph, the identification of the attribute and its definitions based on each type of declared fuzzy data.

By storing and defining fuzzy data, querying and entering data becomes possible because the system can understand how to interpret a cloudy data when it occurs. However, only defining a data is not sufficient for the system to perform satisfactorily its insertion. A fuzzy data is similar to a common data, differing only by not having a similar form to the other related data of this set. Thus, it is proposed to use *key-symbols* to indicate to the system that this is a fuzzy data. For example, when the *"young"* property is inserted into a graph database on a vertex, it can be understood as an exact datum that refers to the characteristic of a person. However, this data may be replacing an exact value unknown at the time, which refers to the age property of a person. An *"young"* in fuzzy logic symbolizes a linguistic term that composes a *linguistic variable*. The use of the key symbols overcomes this problem and can be used for each of the fuzzy data types, such as: *null values*, *sets*, *intervals*, *linguistic variables* e *fuzzy expressions*. The Figure 5 shows the relationship between the key symbols, the proposed statement syntax, and the related data type.

In addition to assigning the key-symbols to the fuzzy data, the *Cypher* language has been modified to simplify the use of these symbols. Thus, a user need not know the symbol of the data type, it should only indicate the syntax of the instruction to be used.



```
Algorithm 1 Fuzzy Data Definition (FDDD.xml)
Input: Cypher instruction
Output: Element created in the file FDDD.xml

Variables:
    x ← Instruction initiated by the term 'DEFINE'

function DEFINITION IMPERFECT DICE(x)

    Check type of statement x                          ▷ Identifies the imperfect data type

    if there is no element of "graph position" then
        creates element for the "graph position"       ▷ ex.: property of the vértice
    end if

    if there is no element of "attribute name" then
        creates element "attribute name"               ▷ ex.: age
    end if

    if there is no element of "imperfect type" then
        creates element "imperfect type"               ▷ ex.: linguistic variable
    end if

    Write definitions of imperfect data.

end function
```

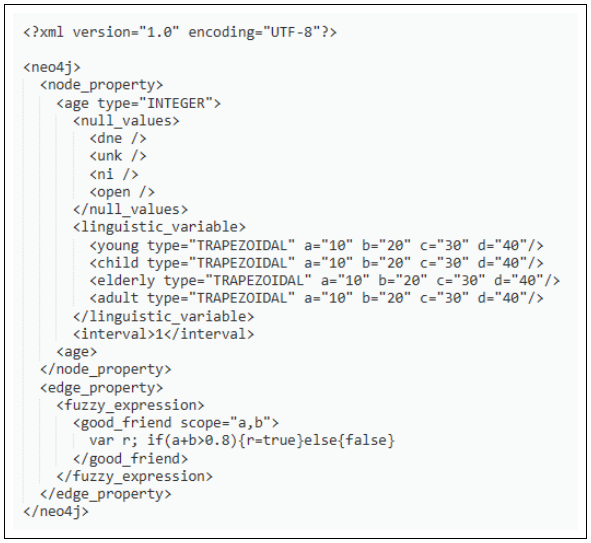
Figure 3: Algorithm for defining the imperfect data in the structure of the *Fuzzy Data Definition Document.*

CREATE (n:Person {name:"Madisara", age IS young})
CREATE (n:Person {name:"Madisara", age:">>>young"})

The conversion between the syntax used and the fuzzy data type is performed internally automatically by the system. This allows the data to be recognized by the system as well as visually by the users.

The query instructions must accept the exact existing data as well as the fuzzy data entered by the previous statement, because, by making use of *FDDD* the query is enabled. Unlike the form used in the insert statement, key-symbols are not assigned to the fuzzy data. The parameters serve to guide the system as to how to interpret the data, both in the instruction and in the result. In this way, we make it possible to execute the following sample instructions.

MATCH (n:Person) WHERE n.age IS 18 RETURN n
or
MATCH (n:Person) WHERE n.age IS young RETURN n
or
MATCH (n:Person) WHERE n.age IS young WITH THRESHOLD 0.7 RETURN n
or
MATCH (n:Person)-[r:FRIEND #GOOD_FRIENDIS(age, gender)]->(m:Person) WHERE n.age IS young WITH THRESHOLD 0.7 RETURN n



```
<?xml version="1.0" encoding="UTF-8"?>

<neo4j>
  <node_property>
    <age type="INTEGER">
      <null_values>
        <dne />
        <unk />
        <ni />
        <open />
      </null_values>
      <linguistic_variable>
        <young type="TRAPEZOIDAL" a="10" b="20" c="30" d="40"/>
        <child type="TRAPEZOIDAL" a="10" b="20" c="30" d="40"/>
        <elderly type="TRAPEZOIDAL" a="10" b="20" c="30" d="40"/>
        <adult type="TRAPEZOIDAL" a="10" b="20" c="30" d="40"/>
      </linguistic_variable>
      <interval>1</interval>
    <age>
  </node_property>
  <edge_property>
    <fuzzy_expression>
      <good_friend scope="a,b">
        var r; if(a+b>0.8){r=true}else{false}
      </good_friend>
    </fuzzy_expression>
  </edge_property>
</neo4j>
```

Figure 4: Example of a *Fuzzy Data Definition Document*

In the first instruction is used *"18"* an crisp operand , however, the operator has been modified to indicate to the system that crisp and fuzzy data will be accepted as an answer. In the second example, a *linguistic variable* is used, in the case *"young"*, which refers to a set of age values of a person. In the third example, an instruction is presented that applies an acceptance threshold. In this the vertices that contain a degree of pertinence of value inferior to *"0,7"* will not be accepted and consequently discarded of the final result. The last example presents a query statement using the previously defined concepts in addition to vertex relationship analysis. In this case, the vertex of type *"Person"* which has at least one relationship with another vertex will be obtained. The *fuzzy expression "Good Friends"* evaluates the properties of the vertices *"age"* e *"gender"* to define the degree of intensity of the relation of these vertices. The degree of intensity defined by a *fuzzy expression* modifies the result of the relation of the vertices. Thus, we know that these vertices are related to a certain intensity.

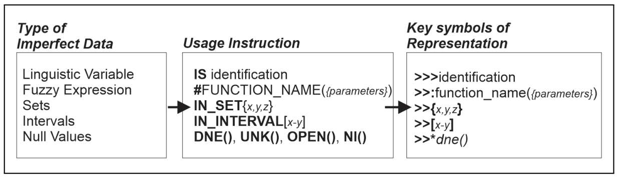
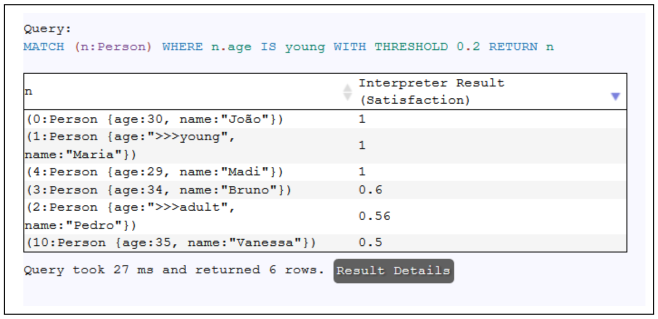


Figure 5: Key-symbols used with the relation of the imperfect data type.

Finally, the application adds to the results the degree of compatibility of the instructions with the fuzzy data obtained, in addition to a final general coefficient. In this the fuzzy data used in the instruction with all the data obtained is compared, or the crisp data used in the instruction with all the data obtained. The coefficients are generated in the range of *0* to *1*, where the closer to *1* the more compatible is the result with the declared statement. In addition to informing the user about the compatibility between the instruction and the results, the coefficients also serve to classify these results. *"Degree of satisfaction"* one can rank among the highest and lowest compatible results.

The Figure 6 presents an example of these concepts.



```
Query:
MATCH (n:Person) WHERE n.age IS young WITH THRESHOLD 0.2 RETURN n
```

n	Interpreter Result (Satisfaction)
(0:Person {age:30, name:"João"})	1
(1:Person {age:">>>young", name:"Maria"})	1
(4:Person {age:29, name:"Madi"})	1
(3:Person {age:34, name:"Bruno"})	0.6
(2:Person {age:">>>adult", name:"Pedro"})	0.56
(10:Person {age:35, name:"Vanessa"})	0.5

Query took 27 ms and returned 6 rows. Result Details

Figure 6: Result obtained by the imperfect query instruction and its degree of satisfaction.

In the proposed application the main concepts of fuzzy logic such as *trapezoidal functions*, *similarity functions* e *fuzzy relation* are incorporated. Each function is used to obtain the degree of pertinence of a fuzzy data that represents the best one. For example, we assign the value *1* when crisp data are used or obtained, since they do not change. In the occurrence of an crisp and a fuzzy value, the membership degree is grasped by the *trapezoidal function*, when applicable. In the occurrence of two fuzzy values, the degree of pertinence is obtained by the *Jaccard's similarity function*. In this way it is possible to measure much of the fuzzy data, applying the definitions based on the interpretation of the user..

5 RESULTS

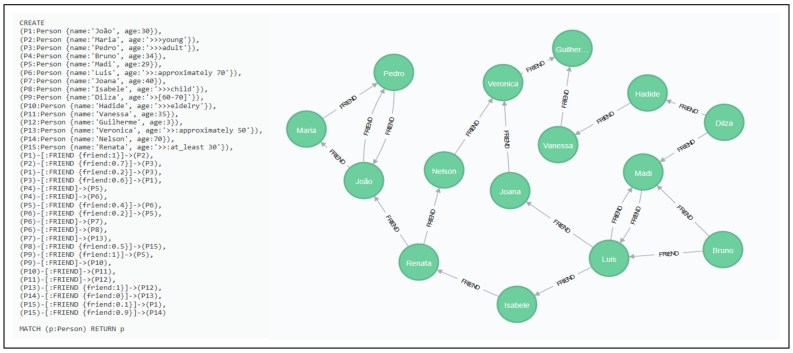
The proposed application integrated to the *Neo4j* system the incorporation of the concepts about fuzzy data used in both the instructions and the analysis of data stored in the graph structure.

It became possible to apply the flexible instructions presented by [3], and to improve the [10] model. Using a separate data structure of the system enables data to be defined, changed, or shared without compromising the integrity of the original system. However, the application of other types of data is necessary to continue the improvement of the model. As a way to validate the proposed application, we will make use of the following use case. As Figure 7 we have an example of a common social network. This is represented by the database of graphs, its vertices represent *Persons* and their edges the relationships existing between them.

Representing the social relationship between people is simple in general, however, if we conduct a deeper analysis, in many cases we will need data that is not fully understood by traditional systems. Based on the proposed social network, we create some situations that validate the fundamental concepts addressed in the proposed application.

- **We will insert a new Person in the network, but we do not know his exact age, only that it has the young appearance.**:

  CREATE (p:Person {name:"Madisara", age:">>>young"})



```
CREATE
(P1:Person {name:'João', age:30}),
(P2:Person {name:'Maria', age:'>>>young'}),
(P3:Person {name:'Pedro', age:'>>>adult'}),
(P4:Person {name:'Bruno', age:34}),
(P5:Person {name:'Madi', age:29}),
(P6:Person {name:'Luis', age:'>>:approximately 70'}),
(P7:Person {name:'Joana', age:40}),
(P8:Person {name:'Isabele', age:'>>>child'}),
(P9:Person {name:'Dilza', age:'>>[60-70]'}),
(P10:Person {name:'Hadide', age:'>>>eldelry'}),
(P11:Person {name:'Vanessa', age:35}),
(P12:Person {name:'Guilherme', age:3}),
(P13:Person {name:'Veronica', age:'>>:approximately 50'}),
(P14:Person {name:'Nelson', age:70}),
(P15:Person {name:'Renata', age:'>>:at_least 30'}),
(P1)-[:FRIEND {friend:1}]->(P2),
(P2)-[:FRIEND {friend:0.7}]->(P3),
(P1)-[:FRIEND {friend:0.2}]->(P3),
(P3)-[:FRIEND {friend:0.6}]->(P1),
(P4)-[:FRIEND]->(P5),
(P4)-[:FRIEND]->(P6),
(P5)-[:FRIEND {friend:0.4}]->(P6),
(P6)-[:FRIEND {friend:0.2}]->(P5),
(P6)-[:FRIEND]->(P7),
(P6)-[:FRIEND]->(P8),
(P7)-[:FRIEND]->(P13),
(P8)-[:FRIEND {friend:0.5}]->(P15),
(P9)-[:FRIEND {friend:1}]->(P5),
(P9)-[:FRIEND]->(P10),
(P10)-[:FRIEND]->(P11),
(P11)-[:FRIEND]->(P12),
(P13)-[:FRIEND {friend:1}]->(P12),
(P14)-[:FRIEND {friend:0}]->(P13),
(P15)-[:FRIEND {friend:0.1}]->(P1),
(P15)-[:FRIEND {friend:0.9}]->(P14)

MATCH (p:Person) RETURN p
```

Figure 7: Structure of a social network in the graph database model

- **What young people exist in this network?**:
  MATCH (p:Person) WHERE p.age IS young RETURN p
- **Increasing the certainty of the previous result we have. . .**:
  MATCH (p:Person) WHERE p.age IS young WITH THRESHOLD 0.7 RETURN p
- **Which young people know the elderly?**:
  MATCH (p1:Person)-[]-(p2:Person) WHERE p1.age IS young AND p2.age IS elderly RETURN p1, p2
- **Which Persons have a "good friend "? Imagining that people of close age and same gender are more likely to have a stronger relationship.**:
  MATCH (p1:Person)-[r:FRIEND #GOOD_FRIENDS(age, gender)]->(p2:Person) RETURN p1, r, p2

All the examples of instructions presented were incorporated by the application proposed in this work and did not exist previously, neither in the traditional model nor in the works already existing in the literature. This information could not be stored in a traditional database model, nor could it be interpreted by its applications. For the most part, this information would be discarded because it would not be considered valid due to its imperfect nature.

It is expected that with the concepts demonstrated in the development of this application, it can be deployed and integrated as one of the functionalities of the *Neo4j* system. However, other validations must be done with different types of fuzzy data in different situations. In addition, it is proposed to develop an algorithm that acts in a generic way among the different types of data. Improved user interface should be better exploited to provide greater support in the definition and use of fuzzy data. It is also indicated the application of artificial intelligence concepts, because with the integration of user preferences, the definitions about the fuzzy data can be automatically defined.

CONCLUSION

This paper presents a proposal for the integration of imperfect data and fuzzy logic using the graph database management system *Neo4j*. The use of that system was motivated for the uses of *RabbitHole* a previous tools for fuzzy data. In that work we introduced new graph database fuzzy data, and also the queries bases on vertex and edges directional paths. With the application developed, a graph can become even more flexible, allowing new types of fuzzy data, at all levels of the graph. The use case based on a social network was implemented and showing the validation of the project. We hope that this project can continue, and is being improved with more imperfect data. In addition it can become a useful tool, contributing to improve the relationship between users and various systems.

In order for the project to continue, we make it available in its current form in the *GitHub*[6], can be accessed through the link:

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

REFERENCES

[1] Renzo Angles. 2012. A comparison of current graph database models. In *Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on*. IEEE, 171–177.

[2] Renzo Angles and Claudio Gutierrez. 2008. Survey of graph database models. *ACM Computing Surveys (CSUR)* 40, 1 (2008), 1.

[3] Arnaud Castelltort and Anne Laurent. 2014. Fuzzy queries over NoSQL graph databases: perspectives for extending the cypher language. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, 384–395.

[4] Rohit Kumar. 2015. Graph databases: A survey. In *Computing, Communication & Automation (ICCCA), 2015 International Conference on*. IEEE, 785–790.

[5] ZM Ma and Li Yan. 2007. Fuzzy XML data modeling with the UML and relational data models. *Data & Knowledge Engineering* 63, 3 (2007), 972–996.

[6] Z. M. Ma. 2007. A Literature Overview of Fuzzy Database Modeling. In *Intelligent Databases: Technologies and Applications*. IGI Global, 167–196.

[7] Witold Pedrycz and Fernando Gomide. 1998. *An introduction to fuzzy sets: analysis and design*. Mit Press.

[8] Raqueline Penteado, Rebeca Schroeder, Diego Hoss, Jaqueline Nande, Ricardo M Maeda, Walmir O Couto, and Carmem S Hara. 2014. Um estudo sobre bancos de dados em grafos nativos. *X ERBD-Escola Regional de Banco de Dados* (2014).

[9] Frederick E Petry. 2012. *Fuzzy databases: principles and applications*. Vol. 5. Springer Science & Business Media.

[10] Olivier Pivert, Olfa Slama, Grégory Smits, and Virginie Thion. 2016. SUGAR: A graph database fuzzy querying system. In *IEEE International Conference on Research Challenges in Information Science (RCISâĂŽ16) Demos*.

[11] Olivier Pivert, Virginie Thion, Hélène Jaudoin, and Grégory Smits. 2014. On a fuzzy algebra for querying graph databases. In *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*. IEEE, 748–755.

[12] Ian Robinson, Jim Webber, and Emil Eifrem. 2013. *Graph databases*. " O'Reilly Media, Inc.".

[13] Marko A Rodriguez and Peter Neubauer. 2012. The graph traversal pattern. In *Graph Data Management: Techniques and Applications*. IGI Global, 29–46.

[14] MS Sunitha and Sunil Mathew. 2013. Fuzzy graph theory: a survey. *Annals of Pure and Applied mathematics* 4, 1 (2013), 92–110.

[15] Lotfi A Zadeh et al. 1965. Fuzzy sets. *Information and control* 8, 3 (1965), 338–353.

[6] *https://github.com*

Climate Change Data Analysis: Case study of Banana in the French West Indies

Nathan Jadoul
LAMIA Lab., University of the French West Indies
France
nathan.jadoul@etu.univ-antilles.fr

Erick Stattner
LAMIA Lab., University of the French West Indies
France
erick.stattner@univ-antilles.fr

ABSTRACT

Last decades, numerous works have been concentrated on climate deregulation. While several studies have analysed the issue on a worldwide scale, few works focused on little land territories. For instance, this is the case of little islands in the Caribbean Sea, for which very few works have been directed to understand the effect of climate deregulation at the nearby dimension. In this work, we center around the French island of Guadeloupe in the French West Indies and we have conducted a study that has two goals. Firstly, we analyse climate information from the previous 50 years regarding feature observers to climate deregulation. Then, we demonstrate the effects of these interruptions on the agricultural area, specifically on the development of bananas broadly cultivated on the island. This methodology, guided by field information, gives a superior comprehension of the difficulties presented by environmental changes in this area and their effects on certain yields touchy to this kind of deregulation.

CCS CONCEPTS

• **Information systems** → **Data management systems**; • **Theory of computation** → *Data structures design and analysis*; • **Applied computing** → *Agriculture*;

KEYWORDS

data science, data analysis, knowledge discovery, climate change

ACM Reference Format:
Nathan Jadoul and Erick Stattner. 2019. Climate Change Data Analysis: Case study of Banana in the French West Indies. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3331076.3331122

1 INTRODUCTION

Last decades, much work has been done on climate change. While some studies have addressed the issue of worldwide climate change [5, 10], few works have been focused on little territories. For instance, this is the case of the small islands of the Caribbean Sea, for which a modest number of studies breaking down the effect of climate change can be found in the litterature [8]. However the impacts of climate change are being felt in these territories [2, 9]. Thus, understanding environmental changes and assessing their potential effects on crops and harvests is an essential issue for the flexibility of the population and the adjustment of rural practices [11].

In this work, we focus on the *Guadeloupe* French Island located in the French West Indies. Unlike the large majority of the works that are interested in climate projection models [7], here we are conducting a data-driven approach that has two goals.
(i) First, we analyse data over the past 50 years to identify evidence of climate change. Indeed, our first objective is to highlight, using several indicators, the climatic deregulation that occurred over past 50 years.
(ii) Then, we seek to understand impacts of the observed disruptions on the agricultural sector, particularly on banana cultivation.

To the best of our knowledge, this is the first data analysis approach that highlights global and seasonal climate trends occurred on the island over the past 50 years. This approach, guided by field data, allows to better understand climate tendencies in this territory, and their impacts on some sensitive crops, like bananas. But they are more curious to know how climate change may evolve and impact banana fields in the future.

2 RELATED WORKS

For decades, the scientific community has been interested in the issue of climate change, but the multiplication of sensors and their evolution has led to the need to manipulate large amounts of data. This is why big data processing methods are generally used in this case. Nevertheless, the use of big data alone does not guarantee a relevant climate analysis because of the many particularities of this paradigm. It is necessary to adapt traditional methods to new issues specific to climate data such as time [6]. But not only big data needs to be reviewed, but also data processing and classification methods must be modified to address the climate problem, such as artificial neural networks or the clustering methods [12, 13], which must evolve to take this time aspect into account in their information processing to be able to capture complex relationships, discover spatial structure and integrate predictive modelling. In addition, it shows that several major ocean climate indices are closely linked to the Earth's climate.

Studies have already been carried out to highlight climate change. Some of these studies study it at a broader level such as continents or the globe [5]. This paper presents the state of knowledge on observed precipitation and attempts to discern some general patterns at the main regional and continental levels that led to the observation of increased variance in precipitation around the globe.

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331122

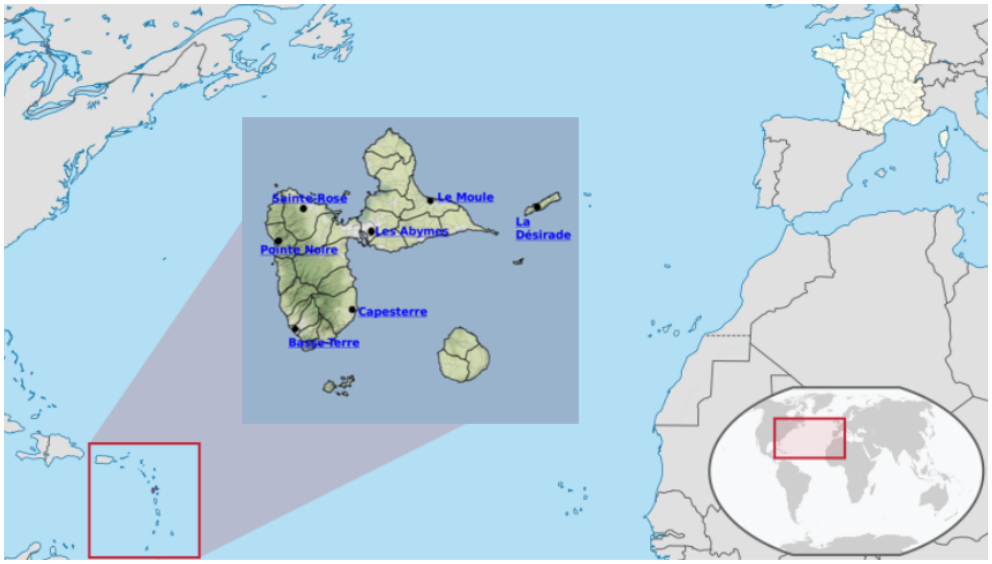


Figure 1: Location of sensors on the island

On the other hand, we have studies that are limited to country levels [1], but which can lead to climate indicators that are relevant for climate change and applicable to other regions of the world, such as the amplitude of daytime temperatures or the monthly average temperature.

Nevertheless, there are still geographical areas where studies on climate change are somewhat behind, either because of its complexity or the lack of general interest, the Caribbean area fits in this case. As a result, there are few studies dealing with climate indicators and these developments in the Caribbean basin [2]. Despite this, studies on the Caribbean Basin have been conducted, from which researchers have come to several conclusions. First of all, the particularity of the land-sea interface very present in the Caribbean zone and the significant influence of major oceanic phenomena (i.e. ENSO, NAO) in the terrestrial climate [9]. Secondly, it has been demonstrated that there is a summer drought phenomenon in the Caribbean basin [3, 8], which is weaker in the East than in the West and which increases over time.

3 METHOLOGY

For this study, we use French weather sensors to collect meteorological data. The sensors used are located in different communes of Guadeloupe, two in Grande-Terre (Le Moule and Les Abymes), four in Basse-Terre (Sainte-Rose, Pointe-Noire, Basse-Terre, Capesterre) and one in La Désirade. We have sensors present on almost all the territory as shown on Figure 1.

Now that we have seen the position of the sensors, let's look at the data itself. First of all, the sensors do not all start at the same time, the oldest ones start in 1964 and the most recent one in 1997. Then, the sensors measure a maximum temperature value (Tmax), a minimum temperature value (Tmin) and finally a precipitation value in millimeters (P). We have chosen to group the data in a single file for the whole territory, and for this purpose, we have averaged the information in relation to the number of sensors. In addition, we determined the average temperature after carrying out operation $\frac{Tmax+Tmin}{2}$, for each beacon and then averaged this value in relation to the number of sensors. Nevertheless, the data are partially incomplete, only on temperatures. Indeed, in the 1990-2000 decades, many data are missing for sometimes whole months, and this, on several sensors, which leads to rather unreliable data. In addition, in the early years (1965 - 1975), temperature data are given by very few sensors. That is why we have preferred to spread them out for temperatures, so all our treatment at the temperature level will start in 1975.

The indicators studied are divided into two categories, those on precipitation and those on temperature. For precipitation, we have the cumulative monthly rainfall in millimeters and the number of rainy days. For temperatures, we have the maximum, minimum, amplitude and number of hot days.

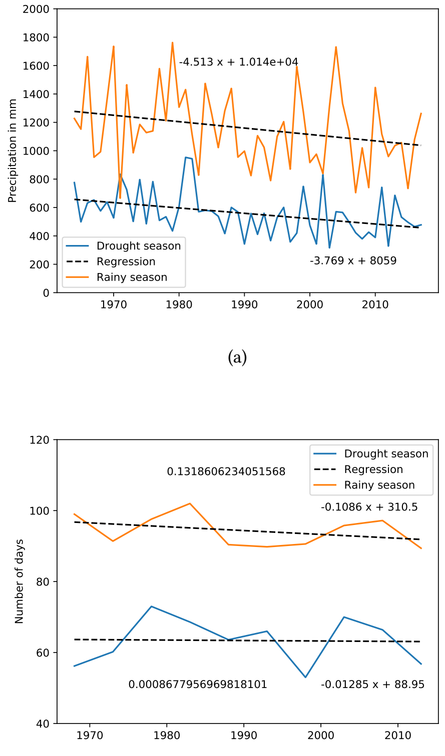
The usual representations for climate data are by month or year. We have decided to proceed with a slightly different approach by looking at these data from a seasonal perspective. In the Caribbean, and more precisely in the West Indies, we have two very different seasons. First of all, Lent, which is the cold and dry season, and Wintering, which is hot and humid. We have considered that Lent extends from December to April and that Wintering extends from June to November. Our objective for this study is to identify indicators of climate change and to see their influences on agriculture. Seasonal representation has allowed us to identify more significant trends than by month or year, these trends will be discussed in the following section.

4 DATA ANALYSIS WORK

The first step in our study was to ensure that several witnesses to climate change stand out from the data. For this purpose, we have extracted different indicators, based on analyses of temperature and precipitation data, which we explore at seasonal paradigm levels

Although the overall analysis above provides a measure of the intensity of the changes that have occurred over the past 50 years, it is unfortunately incomplete in identifying the impact of these changes on agriculture. It is therefore necessary to go through a seasonal paradigm. Indeed, many crops on the island are season-dependent. In this situation, adapting agricultural practices and respecting the plant development cycle requires a deep understanding of seasonal changes. Thus, in the second part of our study, we focused on changes over the island's two seasons, the drought season and the rainy season, in order to understand how climate change affects them.

Figure 2 shows, for both seasons, how (a) the sum of precipitations and (b) the number of rainy days, has evolved since 1965.



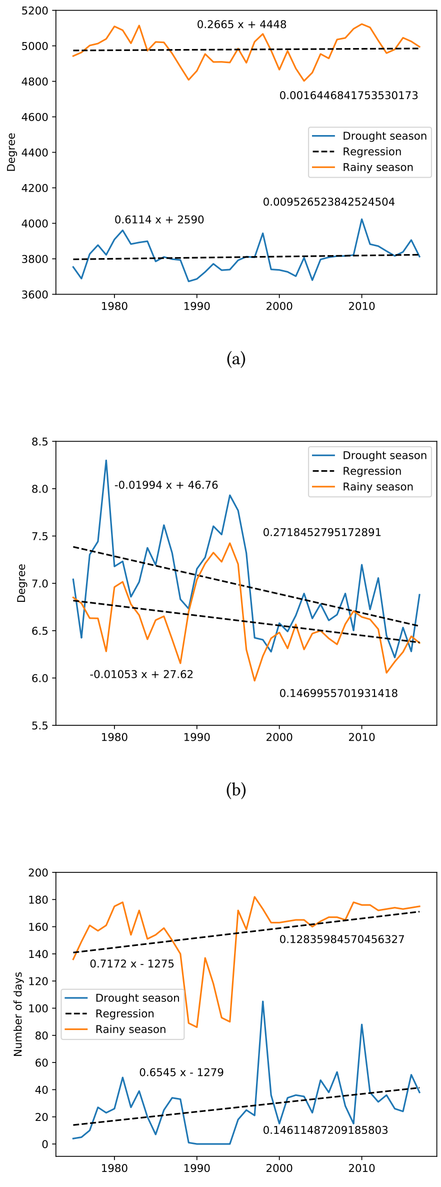
(a)

(b)

Figure 2: For both seasons: evolution of (a) sum of precipitations and (b) number of rainy days

It is normal to note that the general trend is towards a decrease in rainfall, given the results over the years. In addition, this decrease can be observed both in the sum of precipitation and in the number of rainy days. In addition, it can also be observed that the wet season is more affected than the dry season, since the loss of rainfall is greater during the rainy season, as can be seen on the slopes of the regression curves.

Regarding the evolution of temperatures according to the season, we focus on (a) sum of temperatures, (b) thermal amplitude, and (c) the number of hot days. Figure 3 shows the results obtained.



(a)

(b)

(c)

Figure 3: For both seasons: evolution of (a) thermal accumulation, (b) thermal amplitude and (c) number of hot days

First of all, it can be noted that over both seasons, the sum of temperature increases with the years (see Figure 3(a)). It can be noted, however, that the change is much more significant for the dry season.

The results obtained for the thermal amplitude are very interesting (see Figure 3(b)). While the trend is downward for both seasons, the decrease is much greater for the dry season.

If these trends continue, there may be a shift in temperature amplitudes between the two seasons in the coming years. We would therefore have lower temperature amplitudes during the dry season than during the rainy season.

To finish, we focused on the number hot days for both seasons (see Figure 3(c)). For the latter indicator, the trends are very similar to previous observations. This means that an increase can be recorded for the rainy and drought seasons. This confirms the fact that the seasons tend to warm up.

5 CASE STUDY OF BANANA

Agriculture sector is an important part of the economy of the island, since according to the french ministry of agriculture [4], it employs 12% of the active population. Bananas are one of the most cultivated crops on the island of Guadeloupe with about 7.00%.

Banana is a crop sensitive to climatic variations, that directly affect the yields. Thus, in the second phase of our data analysis work, we sought to understand the consequences of the climatic variations highlighted on the plant. To better understand the impact of climate trends on banana, we studied the climate changes in relation to the life cycle of the plant.

The development of banana is closely related to the climate conditions [11]. In our dataset, we only have data on rainfall and temperatures on the past 50 years. However, these two components are essential, since water and heat are favorable for growth. Thus some optimum climate conditions for the plant development are the following:

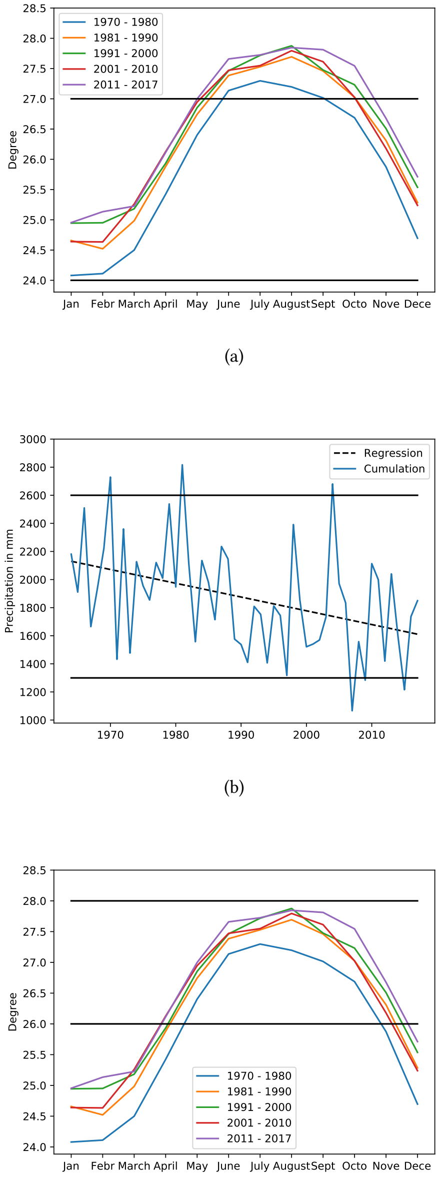
- Growth phase:
  - Temperature: 24°C ≤ temperature ≤ 27°C
  - Rainfall: $1300mm \leq$ annually precipitation $\leq 2600mm$
- Foliar phase:
  - Temperature: 26°C ≤ temperature ≤ 28°C

In this last part of our study, we have analyzed the data in order to track the occurrence of these optimum conditions over the last decades. Figure 4 shows the evolution of the appearance of optimum conditions for banana development according to (a) temperatures for growing, (b) annual precipitation on the last five decades, (c) temperatures for foliar development.

First of all, if we focus on optimum conditions of temperatures for growing phase (see Figure 4(a)), the results are striking. Indeed, it is rather unexpected to see that the conditions favorable to the development of the banana seem to shorten with decades. Thus if the results show that average temperatures increase over the years, this also results in periods of much shorter favorable temperature conditions to the banana development.

With regard to optimal precipitation conditions, very interesting observations can also be made (see Figure 4(b)). Indeed, changes in precipitation seem to tend towards an exit of favorable period at a long term.

If we focus on the optimal temperature conditions for the foliar development phase (see Figure 4(c)) we can observe that the trend is not the same. Indeed, the favorable conditions to the foliar development of the banana seem to lengthen with the decades. The results therefore show a spreading of the optimal period in the year for the foliar development of banana.



(a)

(b)

(c)

Figure 4: Occurrence of optimum conditions for banana development according to (a) temperatures for growing, (b) annual precipitation and (c) temperatures for foliar developement

6 CONCLUSION

In this paper, we have used a data analysis approach to address climate change. Unlike approaches that center on climate modelling for providing projections, here we have conducted a data analytics work to highlight evidences of climate alter extracted from field data. The work we have carried out is centred on the Guadeloupe French Island, located in the French West Indies, and tend to bring out climatic propensities occurred within the past 50 years and their impact on agriculture.

Thus we first have gathered all rainfall and temperature data from various beacons on the island. Then, data were supplemented by several climate indicators, calculated from the data, which are expected to reflect significant changes in climate and impact plant development.

In the second part of the work, we carried out data mining work to highlight the strong climate trends that have occurred over the past 50 years. As seasons are known for their important role in plant development, we then adopted a seasonal approach to highlight changes that occur over the two seasons of the Guadeloupe Island.

Finally, in the last part, we focused on bananas, which is cropped widely cultivated on the island that is known to be sensitive to climatic variations. We have studied how observed climatic trends affect banana development, and more specifically, our approach has made it possible to observe how optimal conditions for banana growth and foliar development have evolved over the decades.

The work we have conducted on this paper opens various interesting research tracks. First of all, in this paper we only concentrated on temperature and rainfall data. However, we can easily suppose that climate change may also be observed on other climate indicators. Thus, at short-term perspectives we plan to complete our dataset by adding other climate data to better characterise changes taking place on global climate and on seasons.

Another interesting approach would be to complete the banana study after adding new climate indicators. Indeed, other climatic indicators involving humidity or sunshine are also known to have an impact on the development of banana trees, and plants in general.

Finally, in a long-term perspective, extracted knowledge could help to adapt agricultural practices in order to have planned development cycles in phase with climate change.

REFERENCES

[1] M Brunetti, L Buffoni, M Maugeri, and T Nanni. 2000. Trends of minimum and maximum daily temperatures in Italy from 1865 to 1996. *Theoretical and Applied Climatology* 66, 1-2 (2000), 49–60.

[2] Philippe Cantet, Michel Déqué, Philippe Palany, and Jean-Louis Maridet. 2014. The importance of using a high-resolution model to study the climate change on small islands: the Lesser Antilles case. *Tellus A: Dynamic Meteorology and Oceanography* 66, 1 (2014), 24065.

[3] S Curtis and Douglas W Gamble. 2008. Regional variations of the Caribbean mid-summer drought. *Theoretical and Applied Climatology* 94, 1-2 (2008), 25–34.

[4] Agriculture Directorate of Food and Forestry. 2010. De la canne Ãă sucre dans la moitiÃľ des exploitation agricoles. *Recensement agricole 2010* (2010).

[5] Mohammed HI Dore. 2005. Climate change and changes in global precipitation patterns: what do we know? *Environment international* 31, 8 (2005), 1167–1181.

[6] James H Faghmous and Vipin Kumar. 2014. A big data guide to understanding climate change: The case for theory-guided data science. *Big data* 2, 3 (2014), 155–163.

[7] Gregory Flato, Jochem Marotzke, Babatunde Abiodun, Pascale Braconnot, S Chan Chou, William Collins, Peter Cox, Fatima Driouech, Seita Emori, Veronika Eyring, et al. 2013. Evaluation of climate models.

[8] Douglas W Gamble and Scott Curtis. 2008. Caribbean precipitation: review, model and prospect. *Progress in Physical Geography* 32, 3 (2008), 265–276.

[9] Isabelle Gouirand, Mark R Jury, and Bernd Sing. 2012. An analysis of low- and high-frequency summer climate variability around the Caribbean Antilles. *Journal of Climate* 25, 11 (2012), 3942–3952.

[10] Gerald A Meehl, Francis Zwiers, Jenni Evans, Thomas Knutson, Linda Mearns, and Peter Whetton. 2000. Trends in extreme weather and climate events: issues related to modeling extremes in projections of future climate change. *Bulletin of the American Meteorological Society* 81, 3 (2000), 427–436.

[11] Julian Ramirez, Andy Jarvis, Inge Van den Bergh, Charles Staver, and David Turner. 2011. Changing climates: effects on growing conditions for banana and plantain (Musa spp.) and possible responses. *Crop adaptation to climate change* 19 (2011), 426–438.

[12] Michael Steinbach, Pang-Ning Tan, Vipin Kumar, Christopher Potter, S Klooster, and Alicia Torregrosa. 2001. Clustering earth science data: Goals, issues and results. In *Proc. of the Fourth KDD Workshop on Mining Scientific Datasets.*

[13] Karsten Steinhaeuser, Nitesh V Chawla, and Auroop R Ganguly. 2010. Complex Networks In Climate Science: Progress, Opportunities And Challenges.. In *CIDU*. 16–26.

DWS: a data placement approach for Smart Grid Ecosystems

Asma ZGOLLI
INPG SA/Grenoble Alps University
Grenoble, France
zgolliasma@gmail.com

Christine COLLET
Grenoble INP/Grenoble Alps University
Grenoble, France

Christophe BOBINEAU
CNRS/Grenoble INP
Grenoble, France
Christophe.Bobineau@grenoble-inp.fr

ABSTRACT

In Smart grid ecosystems, it is important to carefully choose the placement of the datasets across different kind of big data systems in order to achieve high performance of the workloads and conformity with the business and data ecosystem. Our approach for datasets placement is based on metadata about datasets, workloads, and systems. This paper gives a general overview of the data placement module, proposes a high-level design and data model for our solution and presents the placement criteria.

CCS CONCEPTS

• **Information systems → Database management system engines**.

KEYWORDS

datasets, data lakes, big data, optimization, recommendation, metadata

ACM Reference Format:
Asma ZGOLLI, Christine COLLET, and Christophe BOBINEAU. 2019. DWS: a data placement approach for Smart Grid Ecosystems. In *23rd International Database Engineering Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3331076.3331126

1 INTRODUCTION

In their efforts to transform power grids into smart grids, utilities rely on massive deployments of sensors distributed over all components: meters, data concentrators and directly on the power lines, and smart meters at the scale of a country or even a continent. In France Enedis, the major power distribution system operator (DSO) and one of the biggest in Europe, has an on-going plan to deploy more than 35 million smart meters by the year 2021. In this context, more and more data are collected providing fine grain insights about client consumption profiles and the behavior of the power grid. Data are critical towards more efficient management of energy resources and provides new levels of efficiency for business applications. These applications, such as predictive maintenance or demand forecasting, are mainly analytic-based. They are defined by data scientists that do not possess useful knowledge to master the complexity of data applications in a Smart grid ecosystem. A Smart grid big data ecosystem, such as the one proposed by Enedis, relies on a data lake and several project spaces dedicated to user-specific needs. Users have different levels of skills. Users may be data specialists, IT specialists, data scientists, or non-IT professionals. The data lake is based mainly on distributed file systems (such as Hadoop distributed file system HDFS).Data lakes in modern architecture can also include NOSQL stores. In the case of a Smart grid, the lake stores data coming from: (i) smart meters in an Advanced Metering Infrastructure; such data, used for counting, are: public smart meter measurements collected on the distribution grid, quantities of electricity consumed and produced, the consumption and production powers recorded at regular intervals; the maximum powers reached daily and other measures such as reactive energy or average voltage. (ii) sensors distributed over the grid; such data is used for real-time control and monitoring. (iii) outside the electric grid: data from a forecasting center station, pricing catalogs, social media datasets referring to energy and utilities topics, special spatio-temporal data like geographical situations or weather. (iv) customers/electricity distributors: technical data such as the type of meter, the installed power, the existence of special devices for limiting disturbances, etc and classical customer data. As a consequence of the data collection from a wide variety of sources, datasets have different formats, structures, properties, and value distribution. Beyond storage, one of the challenges of a Smart grids ecosystem is to be able to very easily and efficiently process and transform datasets in order to be a support for innovative data processing initiatives. This means that data should be extracted from the lake, loaded in data banks and transformed according to both technical and business requirements. To give an operational dimension to the lake, a data bank is associated with a data bank management system that allows its processing within a so-called project space. A project space includes reliable and efficient workloads that execute SQL-like queries / more basic operations on datasets of the data banks. Workloads may rely on a data bank management system like relational systems such as Teradata [1] , document stores such as MongoDB [2], wide-column stores such as Cassandra [3], key/value stores such as Riak [4], or graph systems such as Neo4J [5]. Workloads can also rely on MapReduce based processing engines like Apache Hive or on MPP based processing engines like Apache Spark [6]. However, these data bank management systems do not guarantee in the same way the performance, latency, scalability, consistency, and availability. From the point of the data lake use, workloads of a project space need to access, join, and process data sets. Defining workloads needs extensive knowledge about data bank management systems. This can be an important issue given that not all users have such skills. As an example of Smart grid ecosystem users, we consider data scientists. Those users in their

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331126

daily work need to access and store datasets, experimentation results... Multi-engine systems such as our ecosystem and its different spaces: the data lake, the data banks, and the data labs, suffer from the data placement problem [7, 8]: (i) where to store datasets (data migration)? ; where to execute the workload (query migration)? ; (ii) what dataset to copy or to move? addressed in some related work as which views to materialize ; (iii) in the case the execution of a data pipeline composed of a set of queries, how to orchestrate the workload execution? what impact on the performance and the scalability? ; (vi) How to minimize the cost of moving a dataset and loading it to a different store? ; (v) How to fragment the data or the query to enhance performance? Our motivation to work on data placement is to address those issues. We aim to design a solution that optimizes the architecture of smart grids data ecosystems in order to minimize the cost of up-front and on-the-fly data transfer and loading between data systems which are generally time-consuming and redundant [9]. This paper presents our approach for effective placement in order to ensure better processing of datasets. A detailed state of the art and a demonstration of the originality of our data placement design was presented in a previous short paper [10]. Our approach is based on the properties of datasets, workloads and the target data systems which are modeled as metadata. We call our approach DWS acronym for Datasets, Workloads, and Systems and it is driven mostly by use cases from Enedis considering their Big Data ecosystem called B4ALL. The remainder of the paper is organized as follows. Section 2 gives a global overview of our datasets placement approach. It presents the data model for the component of the placement module: the datasets, the workloads, and the systems. Section 3 gives an overview of the used metadata. Section 4 presents the data placement module general design and finally, Section 5 concludes the paper and proposes future research directions.

2 USE CASE

Table 1: Statistics about the Experiment's Smart grid datasets

Dataset	Label	Columns nbr	Rows nbr	Data Size
Conso-inf36	Aggregated clients consummation	13	1048576	1,4 M
Conso-sup36	Aggregated industrial clients consummation	14	981101	0,9 M
Family-prof	Profiles families	2	15	16 KB

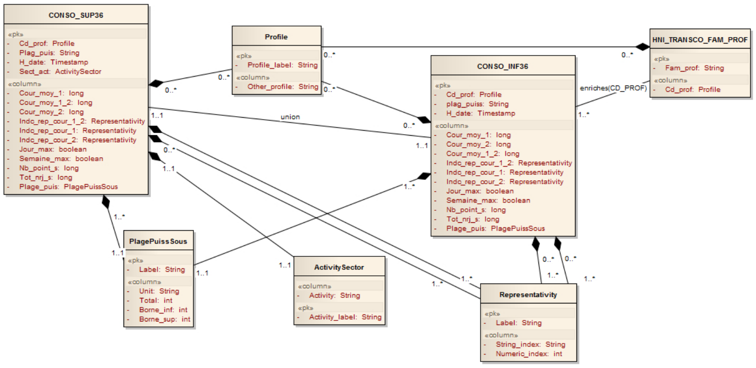


Figure 1: Entity relationship diagram for the experiment's datasets.

The use case of our study is the setup of adapted data infrastructure and optimize the current architecture for Smart grid data ecosystem. The datasets of our experiment are a subset of Enedis's smart grid data. The use case encloses aggregated data about the measure of the client's electricity usage and the table above summarizes and shows statistics about it. This data is characterized by its sensitivity and the current legitimates restricts the research on this kind of data. For this purpose, we are working exclusively on open data that are artificially augmented; Indeed, the original size was multiplied by 6 in order to simulate Smart grids big data ecosystem. For the number of pages limit, this paper will present two example datasets and their associated workload. Those datasets are also characterized by their spatio-temporality, their complex data model and their high volume. The structure of our data is relational and it generally has a snowflake schema. It has a multidimensional data model and it is organized as aggregates produced by analytics applications. Hence, most of the attributes are structured or multivalued. Those datasets reproduce the electricity consumption in a 1/2 h step of the points of withdrawal of electricity connected to the Enedis network and that are less or equal to 36kVA . It gives energy volume withdrawn, the average load curves for customers with smart meters and the number of customers. Average load curve is the average of the volumes of electricity consumed over 1/2 h step given by sites equipped with communicating meters. Average curves are collected for two different categories of consumptions and then aggregated. Enedis also provides for the consumers a curve 's Representativity Index. This attribute is a ratio between the number of points on the Average Curve and the total number of points of the same customer category (same profile, same contracted power range and same industry) Conso-inf36 and Conso-sup36 data sets have similar schemes (cf. figure 1). We present in the following listing the most important columns of those datasets:

- **Horodate** (H-date): a time serie that records the measure's collection time using a half an hour step. It has a DateTime format that represents a point in time defined precisely using the UTC standards. It follows the pattern: YYY-MM-ddTHH:mm:ssZ .
- **Profile** (Profile-label): a categorical attribute that represents a standard profile in the sense of the Recoflux. It can have one of the following values :"PRO" for producers , "ENT3" for a particular type of enterprises,"RES4" for residential customers...
- **Contracted power range** (Plag-puis) : the electrical power subscribed by the user in his supply contract. It represents the maximum possible amount of racking. In these datasets, the different powers are grouped into power ranges. This attribute has complex data type.
- **Number of withdraw points** (Nb-point-s): The number of withdraw points corresponds to the number of sites with an active contract on the Enedis network. It is identical for every 1/2 hour of the same day.

- **Activity sector** (Sect-act): The area of the activity. This attribute only appears in the business consumption data set. The data type of this attribute is complex.
- **Max day of the month** (Jour-max): attribute that indicates whether the 1/2 hours considered is part of the day which reached the peak power consumption of the month in France. the type of this attribute is Boolean (0/1).
- **Week of the month** (Semaine-max): attribute that indicates whether the 1/2 hours considered part of the day which reached the peak power consumption of the week in France. the type of this attribute is Boolean (0/1).
- **Total energy withdrawn (Wh)** (Tot-nrj-s): The total energy withdrawn corresponds to the volume of electricity consumed over 1/2 h given by all the sites of the profile and the power range considered. this attribute has numeric data type with double precision.
- **Average curve C1 (Wh)** (Cour-moy-1): average load curve for of the group of measures whose ratio (Conso 8h-20h)/(Conso total) is the highest. In a like manner, **Average curve C2 (Wh)** (Cour-moy-2) represents the group with the lowest ratio (Conso 8h-20h)/(Conso total) and **Average curve C1 + C2 (Wh)** (Cour-moy-1-2) is the average for all sites.
- **Curve's Representativity Index C1** (Indc-rep-cour-1): representativity index of curve 1 expressed in percentage. Similarly, two other columns are defined : **Curve's Representativity Index C1 + C2** and **Curve's Representativity Index C2** for the representativity index of the other types of curves.

In smart grid data ecosystem, analytics applications transform the datasets using the sequel query language. Those applications have multiples joins, compute several aggregations and contain specific OLAP query operators.For our example, we choose a workflow that covers in the same time join operators, aggregations and temporal operators. The examples represented in this paper are extracted from an application that enriches clients' consummation data with a normalized value of the electric power.

3 OVERVIEW

In this section, we begin with providing a high-level overview of the datasets placement in smart grid ecosystems. We then briefly present our approach that captures the intermediate representation of a query workload as a graph of operators and utilizes the data sets properties and the data processing systems descriptions to achieve a data placement that minimizes the cost of the workload execution. The placement process may take place into separate phases of a workload: just before the processing, on the input datasets, or after the processing, on the result and intermediate result datasets. Our data placement process takes as input a dataset S = <di,.., dn> and a workload W containing query operation <op1, ... opn> on S. Based on the properties of S and the data operations done on S, the placement module chooses the most adequate data system(s) for storing S and for processing S to guarantee efficient performance of W.A potential data placement solution can be: the storage and workload execution by a document data store like Mongodb or the storage in the distributed file system HDFS and the workload execution by a parallel processing engine like Spark. We seek to find the most effective solutions. In the following subsections, we detail the data model of the inputs (the dataset S and workload W).

3.1 Datasets model

For the internal representation of the datasets in our placement module, we adopt the ODMG [11] data model. This model proposes two constructors to organize the values into datasets: the collection constructor that organizes values into sets, lists, bags, arrays, and dictionaries, and the aggregation constructor that organizes the values into objects or aggregates.In this model, values vi are either literals or aggregates. Those values are also categorized into atomic, structured or collections. We represent null values and unstructured values (e.g binary large objects: blobs) respectively as the Null value and an aggregated value. To build a dataset, we first use the collection constructor. It groups related aggregates di in a collection S:

S= <di,..,dn>.

The aggregate constructor creates objects that are either structured (e.g. a tuple of a relational dataset), semi structured (e.g. a document) or unstructured (e.g. a blob). In our model, we keep the same structure for the aggregates and we represent the absence of an attribute-value pair as a missing value. Hence, missing values and null values are two different representations in our model, we associate to each a different literal. The variety of the data manipulated in our ecosystem motivates us to consider this flexible model. With this representation, we can process datasets managed in data stores having a relational data model, key-value store, document store or a graph based store. For instance, for a relational table, we represent the relations with the collection constructor and the tuples with the aggregate constructor. In document stores, aggregates are represented as documents and organized in collections. We represent attribute-graphs by two datasets: nodes datasets and edges dataset. We use two separate collection constructors for the nodes dataset and the edges dataset. Aggregate constructors are also used to represent nodes and edges. Finally, we represent datasets stored in key-value stores with the collection constructor and key-value pairs with an aggregate (a unary aggregate).

3.2 Workload model

A workload in big data ecosystem quantifies the processing performed on data systems. In our context, we consider query workloads derived from SQL applications and scripts used in the consumption aggregation use case. According to the proposed model, query workloads are grouped as flows of operations (or activities) and they are executed sequentially in data pipelines. Thus, we represent a query workload as a graph G= <V,E>. V is the set of vertices representing a logical operator, a variable or a value and E is the set of edges representing dependencies between the vertices. In the

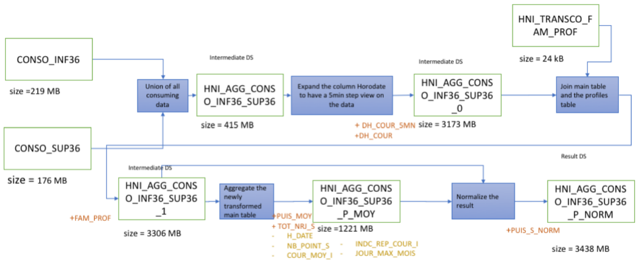
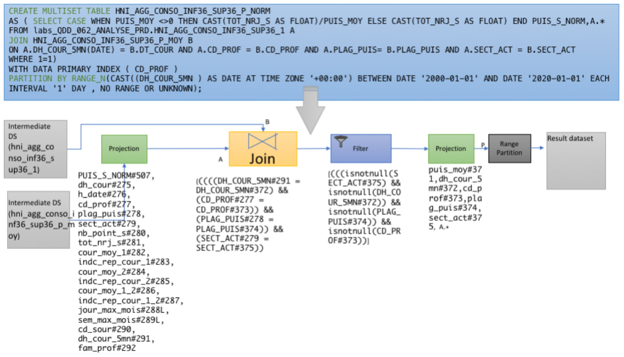


Figure 2: Schema of a data pipeline in a smart grid open data application use case.

query workloads graph, we use the intermediate representation of the query workload for the data activity structure. This intermediate representation can be an asymmetric syntax tree (AST), e.g. Mongodb access API workloads, or a directed acyclic graph (DAG). The choice of representing the workload using the intermediate representation of the queries facilitates the execution of the placement algorithm independently to the environment and the target query language. In our experiment and for simplification reasons, we use ASTs. Figure 2 presents an example of a compact view data pipeline of a smart grids data processing application. This pipeline is structured as a DAG of data activity. Smart grid data activities generally aggregate different datasets. They usually scale out using horizontal partitioning and apply many filters and transformations to obtain the desired output. As an example of a data activity, we consider the following graph (figure 3). This activity is the first step of the pipeline. It enriches the perimeter dataset with the date of the calculation of the indicator and stores the result in an intermediate table that will be used as an input for the remaining of the pipeline.



```
CREATE MULTISET TABLE HNI_AGG_CONSO_INF36_SUP36_P_NORM
AS ( SELECT CASE WHEN PUIS_MOY <>0 THEN CAST(TOT_NRJ_S AS FLOAT)/PUIS_MOY ELSE CAST(TOT_NRJ_S AS FLOAT) END PUIS_S_NORM,A.*
FROM labs_QDD_062_ANALYSE_PRD.HNI_AGG_CONSO_INF36_SUP36_1 A
JOIN HNI_AGG_CONSO_INF36_SUP36_P_MOY B
ON A.DH_COUR_5MN(DATE) = B.DT_COUR AND A.CD_PROF = B.CD_PROF AND A.PLAG_PUIS= B.PLAG_PUIS AND A.SECT_ACT = B.SECT_ACT
WHERE 1=1)
WITH DATA PRIMARY INDEX ( CD_PROF )
PARTITION BY RANGE_N(CAST((DH_COUR_5MN ) AS DATE AT TIME ZONE '+00:00') BETWEEN DATE '2000-01-01' AND DATE '2020-01-01' EACH
INTERVAL '1' DAY , NO RANGE OR UNKNOWN);
```

Intermediate DS (hni_agg_conso_inf36_sup36_1)
Intermediate DS (hni_agg_conso_inf36_sup36_p_moy)
Projection
B
A
Join
Filter
Projection
P
Range Partition
Result dataset
PUIS_S_NORM#507, dh_cour#275, h_date#276, cd_prof#277, plag_puis#278, sect_act#279, nb_point_s#280, tot_nrj_s#281, cour_moy_1#282, indc_rep_cour_1#283, cour_moy_2#284, indc_rep_cour_2#285, cour_moy_1_2#286, indc_rep_cour_1_2#287, jour_max_mois#288L, sem_max_mois#289L, cd_sour#290, dh_cour_5mn#291, fam_prof#292
((((DH_COUR_5MN#291 = DH_COUR_5MN#372) && (CD_PROF#277 = CD_PROF#373)) && (PLAG_PUIS#278 = PLAG_PUIS#374)) && (SECT_ACT#279 = SECT_ACT#375))
[(((isnotnull(SECT_ACT#375) && isnotnull(DH_COUR_5MN#372)) && isnotnull(PLAG_PUIS#374)) && isnotnull(CD_PROF#373))]
puis_moy#371,dh_cour_5mn#372,cd_prof#373,plag_puis#374, sect_act#375, A.*

Figure 3: Example of a data activity in a smart grid data processing application.

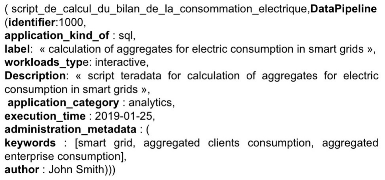
3.3 Systems model

A data system in big data ecosystems is either a data storage system like data stores, databases or distributed file systems (DFS) or a processing engine, e.g MPP based processing engines. We model for systems 2 entities: (i) Abstraction of a data system that details the characteristics of those systems modeled from state of the art surveys and experimentation on those systems. And (ii) Systems descriptors that represent available data systems in the smart grid data ecosystem. We represent a system descriptor as a composite object. We associate to this object two other objects: A Storage Descriptor and a Processing Descriptor. The Storage Descriptor is an object that defines attributes such as the data model, partitioning model, distribution model ... Similarly, the Processing Descriptor has as attribute: supported query and access APIs, physical operators, data model... Descriptors are represented according to the ODMG model as classes; the relationships between those classes are either composition and association or inheritance.

4 METADATA SPECIFICATION

The metadata we use consider three abstract independent but complementary layers of the ecosystem referred to as Applications and Workloads, Data and Systems [10]. Each layer is described by its specific metadata schema.

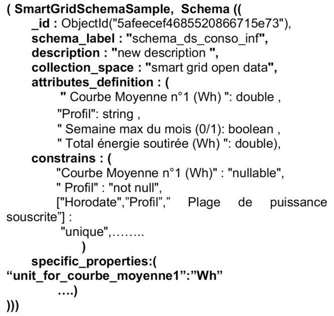
Applications/workloads metadata characterizes query models, query logical operators as well as query workloads and query statistics. Other application metadata considered are semantic annotations and rules.



(script_de_calcul_du_bilan_de_la_consommation_electrique,**DataPipeline**
(**identifier**:1000,
application_kind_of : sql,
label: « calculation of aggregates for electric consumption in smart grids »,
workloads_type: interactive,
Description: « script teradata for calculation of aggregates for electric consumption in smart grids »,
application_category : analytics,
execution_time : 2019-01-25,
administration_metadata : (
keywords : [smart grid, aggregated clients consumption, aggregated enterprise consumption],
author : John Smith)))

Figure 4: Examples of workloads metadata for characterizing a Smart grid data pipeline.

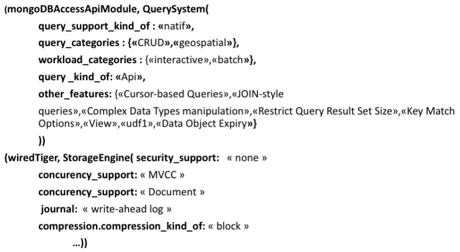
The **datasets metadata** (cf. datasets modeling) describes for instance (but not limited to): dataset size, values distribution, availability, location, schemas, administration metadataâĂę



```
( SmartGridSchemaSample,  Schema ((
      _id : ObjectId("5afeecef4685520866715e73"),
      schema_label : "schema_ds_conso_inf",
      description : "new description ",
      collection_space : "smart grid open data",
      attributes_definition : (
             " Courbe Moyenne n°1 (Wh) ": double ,
            "Profil": string ,
            " Semaine max du mois (0/1): boolean ,
            " Total énergie soutirée (Wh) ": double),
      constrains : (
            "Courbe Moyenne n°1 (Wh)" : "nullable",
            " Profil" : "not null",
            ["Horodate","Profil"," Plage de puissance
souscrite"] :
             "unique",……..
                  )
      specific_properties:(
"unit_for_courbe_moyenne1":"Wh"
            ….)
)))
```

Figure 5: Example of datasets structural metadata.

We characterize datasets with descriptive statistics about the attributes and the records and with structural metadata by keeping track of their local and global schema. An example of a local schema for the dataset conso-inf36 is illustrated in figure 5. We also characterize for semi-structured datasets additional metadata like: the probability of having missing values for an attribute, nesting degree of structured values and structured values cardinality.



```
(mongoDBAccessApiModule, QuerySystem(
        query_support_kind_of : «natif»,
        query_categories : {«CRUD»,«geospatial»},
        workload_categories : {«interactive»,«batch»},
        query _kind_of: «Api»,
        other_features: {«Cursor-based Queries»,«JOIN-style
        queries»,«Complex Data Types manipulation»,«Restrict Query Result Set Size»,«Key Match
        Options»,«View»,«udf1»,«Data Object Expiry»}
        ))
(wiredTiger, StorageEngine( security_support:  « none »
        concurency_support: « MVCC »
        concurency_support: « Document »
         journal:  « write-ahead log »
        compression.compression_kind_of: « block »
                ...))
```

Figure 6: Example of systems metadata about the storage engine of Mongodb and its query system.

The **systems metadata** describe data stores, distributed file systems (DFS) and processing engines that take part of the big data ecosystem. The metadata schema of this level has the following main concepts: partitioning models, data models, query models and storage models for those systems.In our approach, we classified data systems properties. Then according to our classification / taxonomy, we specify additional properties as attributes (Fig 6).

5 DATA PLACEMENT MODULE DESIGN

Let us consider a dataset S as a set of tuples: <d1,...,dn> and a query Q composed of a set of several operations <op1,...,opn> (for instance: select/project/join ...). A data system DS is an appropriate placement for S, if the layout of the dataset is supported by the storage engine and its processing engine supports the query Q and provides an efficient workload execution for Q.

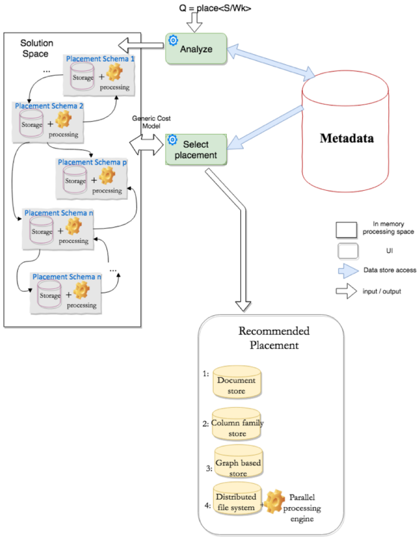


Figure 7: Workflow of a data placement module.

Our placement approach aims to address a set of issues described as follow: (i) big data store has APIs that are too specific to every data model and query workloads. Those systems no longer use the one size fits all solution and their APIs have partially overlapping querying capabilities [9]. Thereby the choice of the storage and the processing systems needs to take this aspect into consideration to ensure fitting the applications need. (ii) the result of the placement can be unacceptable in a productions environment. Allowing the user to freely choose the data system for his need can lead to placement errors. (iii) Sometimes, in the case of cross-referencing two or more datasets, it is more advisable to use native (system level) execution engine for this type of workloads over application level hybrid processing. Indeed, existent parallel DBMS and multi-store engines load the datasets in memory and then execute the query application. This solution creates overhead in the query performance. We design our placement module to help the users to efficiently build they data applications. Datasets placement is identified based on systems characteristics and the functionalities offered by their APIs. In this objective, we propose to consider 3 decision criteria for the placement explained in details in our previous work [10]. We evaluate the **feasibility** of the placement by comparing the characteristics of the target systems. We check the **conformity** of the placement with the data and the business ecosystem using a set of rules. Finally, we estimate the **performance** of query execution using cost models considering different datasets and query transformations. DWS cost model covers query execution , data transformation and data communication between stores. This solution helps us minimize the execution time and the data transfer time.The mechanism behind finding the optimal placement solution is inspired by query evaluation techniques in multi-store systems: As shown in figure 7, the placement process first generates a placement solution space by inferring on metadata. This step decomposes the workload according to the specification provided by the application and represented as a DAG (cf subsection 3.2). Then the algorithm decomposes the queries of the workload into as set of sub-queries and solves the data placement problem for each sub-query combination. In the second step, DWS's placement algorithm identifies the systems that matches the feasibility of the placement as well as its compliance with business environment. Subsequently, this algorithm selects effective placement candidates based on the impact of the query workload and produces the placement schemas to be returned to the user. The final steps of the placement consist of generating the storage and execution configuration then executing the placement. In spite of the importance of the final step, we limit our contribution to presenting the selected placement results to the user as a recommendation.

6 CONCLUSION AND PERSPECTIVES

In this paper, we presented our data placement approach that aims to assist the users in managing the complex smart grid data ecosystem and highlights the purpose and the importance of managing systems level metadata. The systems level metadata that we defined are characteristics of their design and internal architecture. Modeling run-time and configuration properties of those systems can be considered as perspective to this work. For future works on this project, we will include the implementation of our data placement approach and the experimental evaluation of the placement module.

7 ACKNOWLEDGMENTS

This work has been funded through the Industrial Research Chair Enedis.It has been carried out in collaboration with the Enedis Information Systems Division (Franck Atgie, Stephanie DELAUNAY, Sebastien VERDIER TOUREAU)

REFERENCES

[1] Teradata. Available at https://www.teradata.fr/ , Accessed: 2018-05-28.
[2] MongoDB. Mongodb documentation. Available at https://docs.mongodb.com , Accessed: 2017-11-22, 2017.
[3] Apache cassandra, manage massive amounts of data, fast, without losing sleep. Available at http://cassandra.apache.org/ , Accessed: 2018-05-28.
[4] Riak kv. Available at http://basho.com/products/riak-kv/ , Accessed: 2018-05-28.
[5] Neo4J team. The neo4j graph platform - the 1 platform for connected data. Available at https://neo4j.com/ , Accessed: 2018-05-28, Mai 2018.
[6] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. Apache spark: A unified engine for big data processing. *Commun. ACM*, 59(11):56–65, October 2016.
[7] Kunyu Zhang Yang Yue Qing Zhao, Congcong Xiong and Jucheng Yang. A data placement algorithm for data intensive applications in cloud. volume 9, pages 145–156, February 2016.
[8] Rajashekhar M. Arasanal and Daanish U. Rumani. Improving mapreduce performance through complexity and performance based data placement in heterogeneous hadoop clusters. In Chittaranjan Hota and Pradip K. Srimani, editors, *Distributed Computing and Internet Technology*, pages 115–125, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
[9] Jeff LeFevre, Jagan Sankaranarayanan, Hakan Hacigumus, Junichi Tatemura, Neoklis Polyzotis, and Michael J. Carey. Miso: Souping up big data query processing with a multistore system. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 1591–1602, New York, NY, USA, 2014. ACM.
[10] Asma Zgolli, Christine Collet, and Houssem-Eddine Chihoub. Metadata based datasets placement in smart grids. October 2018. article de 4 pages présenté sous forme de poster à la conference internationale MTSR 2018 (cyprus).
[11] R. G.G. Cattell, Douglas K. Barry, Mark Berler, Jeff Eastman, David Jordan, Craig Russell, Olaf Schadowa, Torsten Stanienda, and Fernando Velez. *The Object Data Management Standard: ODMG 3.0*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

On Approximate Nesting of Multiple Social Network Graphs: a preliminary study

Giacomo Bergami
Dept. of CISE
University of Florida, USA
gbergami@ufl.edu

Flavio Bertini
Dept. of Physics and Astronomy
University of Bologna, Italy
flavio.bertini2@unibo.it

Danilo Montesi
Dept. of CSE
University of Bologna, Italy
danilo.montesi@unibo.it

ABSTRACT

A fundamental problem in Social Network Analysis is how to move from single-layer to multi-layer, which provide a holistic view. User profiles resolution has received considerable attention since it allows to match users on different online social networks (OSNs). However, to the best of our knowledge, no study has focused on nesting operation for merging OSNs graphs. This work is a first step in the direction of defining the data model and the algorithm to perform approximate nesting of multiple OSNs graphs, based on user features. We provide initial experimental evidence based on synthetic data.

CCS CONCEPTS

• **Information systems** → **Query optimization**; **Graph-based database models**; • **Networks** → *Online social networks*; • **Security and privacy** → Social network security and privacy;

KEYWORDS

Graph nesting, Approximate graph query answering, Multilayer social networks

ACM Reference Format:
Giacomo Bergami, Flavio Bertini, and Danilo Montesi. 2019. On Approximate Nesting of Multiple Social Network Graphs: a preliminary study. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3331076.3331081

1 INTRODUCTION

In Social Network Analysis an online social network (OSN) is a graph $G = \langle N, E\rangle$ where the nodes N represent users and the edges E represent social connections between them, like friendship, shared interests and working affiliation (Figure 1a). Traditionally, OSNs are studied as separate single-layer graphs. Recently, researchers have come to a holistic vision that includes more than one network at a time, that is *multilayer social networks* [8]. Formally, a multi-layer network $G = \langle N, E, L\rangle$ consists of N nodes (i.e., the users), E edges (i.e., the social connections within and across layers) and L layers (i.e., the different OSNs), as shown in Figure 1b.

The multi-layer perspective allows studying a growing number of social structures and phenomena, exploiting the resultant network enriched by new edges across layers. In particular, the red edges in Figure 1b, namely the *inter-layer edges*, might be produced by a user profiles resolution algorithm identifying which users might be different *personae* impersonated by the same individual. Lately, the nested perspective also gained popularity within the Complex Network community [10] for the ability in representing the interaction between different subcomponents. Nested graphs provide a possible representation for user profiles resolution operation on multi-layered OSNs.

Problem Statement. The resolution of user profiles among different OSNs allows moving from single-layer to multi-layer networks, forming a resultant network (the bottom one in Figure 1b). For the use case in Figure 1b, this task aims at merging (e.g., matching) user profiles on different OSNs in a similar way to what happens with a join operation between tables in a relational database. The resolution of user profiles is of the utmost importance since it allows to create the *inter-layer edges* across the layers. However, because user profiles that belong to the same user can have different user ids, email or nicknames, such resolution is very challenging [4]. Current literature provides different approaches for user profiles resolution using different types of information [17], such as basic user features (e.g., name, user id, mail address) [4, 14], user's activities log (e.g., texting, sharing, reacting) [1, 13], and SN's topological information (e.g., friends and mutual friends) [15].

Proposed Approach. In this paper, we propose an approximate nesting approach for multiple social network graphs using the sensor pattern noise of the images captured and shared through the

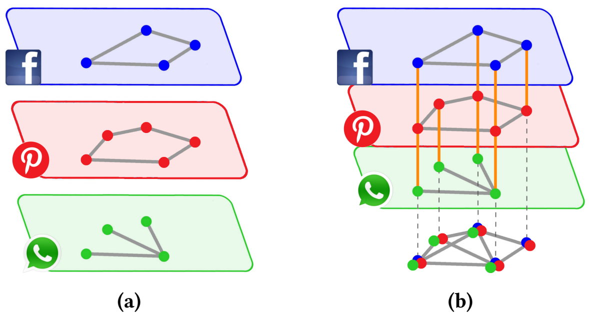

(a) (b)

Figure 1: Single-layers vs multi-layer social networks: three different SNs as separated layers (a); the same three SNs forming the resultant network after the resolution of user profiles (b).

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331081

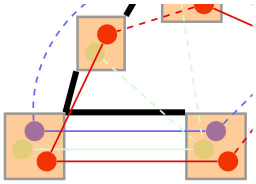


Figure 2: A zoom in of three clusters in the nested graph providing the detailed description of the flattened graph in Figure 1b.

user's smartphone as user profiles resolution technique. In particular, we exploit the clustering approach to solve user profiles resolution and then the graph nesting operator to approximately nesting multiple social network graphs into one single nested graph. For the first approach, we showed in [16] that the method successfully clusters users based on the different cameras that the user exploited for sharing photos in different OSNs. Thus, we exploit our previous work to merge all together users on different OSNs while preserving their information content. For the second approach, we represent each single resulting merged node in Figure 1b as one single node containing all the users matched by the same *inter-layer edge*. Since the same user might use different cameras, a clustering algorithm might put him/her in different clusters, and therefore we might have distinct chains of *inter-layer edges* connecting the same user of the same layer. Therefore, the resulting user resolution is imprecise, and we need that the node nesting allows overlapping containments, that is the same user might appear in a different merged node. The aforementioned approximated nested graph data model also permits non-exclusive (i.e., overlapping) nestings as required by this scenario. We assume that the clustering algorithm flattens the multi-layers social networks as one resulting network, and all the resulting connected components are enriched by drawing an edge between each photo and the cluster to which such photo belongs. The zoomed-in output for the leftmost node clusters in Figure 1b is provided in Figure 2 as a nested graph: the nested nodes containing the nodes connected by the *inter-layer edges* are represented as orange squares, while the nested edges containing the `follows` edges among clusters are represented as bold edges between two squares.

Contribution. Compared to our previous work [7], we need to generalise only the algorithm because the vertex and the edge grouping references are now separated by a distance greater than two (i.e., five). We also propose an alternative graph nesting query plan for this new scenario[1]. The output of this new query plan is yet another nested graph where all the nested nodes contain all the users whose photos belong to the same cluster, and each nested edge connecting two nested nodes contains all the friendship relationships among different layers relating the users to the source cluster to all the others in the target cluster.

2 RELATED WORK

Different approaches have been proposed for user profiles resolution [17]. Works like [4] and [14] exploit information about users' identities, such as usernames, passwords, login information, to match profiles across SNs. In [1] and [13], the authors collect log files within the user's device and study usage patterns to identify user activities on SN and compare common behaviours across SNs. A machine learning technique based on SN topological based features is proposed in [15]. On the flip side, smartphones have several built-in sensors that can be used for identification of devices [3]. In particular, the sensor pattern noise (SPN) is a reliable solution for fingerprinting smartphones due to the imperfections created during their manufacturing process [12] and helps to solve the user profiles resolution problem [16].

The nesting operator η [7] uses two subgraph clustering UDFs g_V and g_E, which are overlapping and partial, to summarize each subgraph into either a nested vertex or a nested edge of the final nested graph. Such operator generalizes the previously defined *graph summarization* and *graph joins* in literature. With respect to *graph summarization*, graph nesting is still more general than other recently proposed graph nesting operations such as [11], given that the presented operator is only able to list set of vertices in nested graphs and not entire subgraphs; in order to obtain that, this last approach is not able to nest entire subgraphs within the final edges. While previous graph summarisation operation mainly provides a partitioning of the graph, this operator allows fuzzy clustering with outliers. *Graph θ-join* [6] ingests two graph operands and returns a single graph, where it both fuses each θ-matched vertex pair from the two distinct components into one single vertex, and creates an edge between each fused vertex accordingly to a specific "edge semantic". If we represent both operands as distinct connected components of one single graph [6], and we use θ to define g_V and the edge semantics to define g_E, the graph join operator is a specific case of the graph nesting operator where, on the other hand, the former loses the provenance information. Last, [7] showed that a two-hop might be expressed in other query languages (both in relational, document-oriented and graph databases) as multiple group-by operations. Given that all such languages do not allow to perform multiple group-by operations simultaneously, their associated query plan shows to be inefficient.

3 LOGICAL AND PHYSICAL DATA MODELS

Given that multi-layer network might be represented as graph collections and given that both a graph database and graph collections might be represented as single distinct connected components of a single graph [6], we choose to represent multi-layer network as one nested graph. This assumption also helps us representing the photo's clustering information in no layer and as nodes shared among different possible social networks.

The distinction between logical (nested graphs) and physical data model is required for distinguishing several roles that the data structures play. First, we represent nested graph operands after the loading and indexing phase as an *extended adjacency list* where each vertex v is associated to its id, its hash, its label-set and the attributes (i.e., properties) and their associated values. The graph is initially created in primary memory without the offset information (*loading*) and afterwards serialised into secondary memory using a specific vertex order detected in the previous phase (*indexing*).

Second, the *nested query result* is only used by the user to read the outcome of the nesting process as in other query languages (such

[1] FoSP source code is available at https://rebrand.ly/FHoSP.

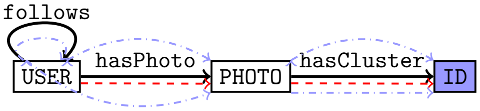


Figure 3: Graph schema of the flattened social networks (thick edges). The user will formulate both the vertex (dashed) and the edge (dash dotted) summarization patterns.

as SPARQL and SQL) and does not have to produce "materialised views". Therefore, the result of the graph query itself can postpone the creation of a complete "materialised view". Such query result represents the adjacency list associated with the resulting nested graph, alongside with a nesting index.

4 FHOSP ALGORITHM

For representing Figure 1b as a nested graph, the user might use the clustering outcome to nest the networks: the user needs the nesting operator requiring two distinct graph patterns, one for nesting the graphs inside the nested vertices (*vertex summarization*), and the other for nesting the graphs inside the nested edges (*edge summarization*). These can be also derived from the graph schema (Figure 3); those patterns are expressed by the following information need: "*After nesting each* USER *into each photo cluster* ID *for each posted* PHOTO*, establish an edge between two photo cluster* ID *if and only if there are two* USER*s which are* followers *and which have photos belonging to those clusters, and nest the original following information within this nested edge*". Instead of traversing all the patterns and then joining them together, we might visit first the PHOTOs to associate each user to all its potential cluster IDs, representing the final nested vertices. Then, we will visit all the follow edges to generate several nested edges connecting the cluster ID.

4.1 Loading and Indexing

Given that each user is identified by the set of the associated photo cluster descriptors, and given that we want to return a nested graph where *(i)* each nested vertex contains all the users having photos associated to the same cluster id and *(ii)* each nested edge between cluster c_i and c_j contains all the following relationships associating users from cluster c_i to the ones in cluster c_j, we want to visit the graph such that first we recognize all the photos and the users belonging to the cluster c_j, and last visiting all the photos and users belonging to the cluster c_i. This visit strategy allows minimising the graph visits required to generate the edge $c_i \to c_j$ and all of its nestings. This problem reduces to find all the possible dependencies within one single multi-layered flattened operand and to visit the graph's vertices in increasing order of mutual dependencies.

In order to validate such assumption, we define three distinct ordering strategies influencing the graph visiting order: ι, $\hbar$ and τ. The first two serialise the operand not taking into account the order of the mutual dependencies, while the last one considers the previous assumption. ι is a loading and indexing strategy that orders the vertices by their ids and serialises the operand's adjacency list accordingly. Given that in our merged layer dataset the vertices' id is randomly assigned for each flattened multi-layered graph g, the visit of g using such order does not necessarily guarantee an optimal ordering. This strategy provides the baseline for the following loading and indexing strategy meeting the requirements stated in the previous paragraph. The hashing strategy $\hbar$ is the same adopted in [7]: we serialise the vertices ordered by label's information and, in particular, we serialise first the cluster ids, then the photos and, last, the users. No specific order is preferred among all the nodes having the same label information. Finally, the topological strategy τ requires a topological ordering of the operand itself. Nevertheless, such ordering requires such operand to be a Direct Acyclic Graph (DAG) while OSN's follow relationships in g cannot generally guarantee such condition. Therefore, we need to detect a *feedback arc set* [19] to break those cycles in the operand g, provide the topological ordering for each resulting DAG, and then serialise the two DAGs as one single graph. Given that finding the minimal *feedback arc set* is an NP-Complete problem, we use the polynomial heuristic defined in [18] to approximate our problem: we generate two DAGs g_1 and g_2 containing the edges (u, v) where $u \leq v$, and those where $u > v$, respectively. Last, the vertices are serialised using the topological order of g_1 and then g_2. As a result, all the indexing costs are linear with the respect to the data size and, in particular, the topological loading and indexing requires an additional linear time to split the operands into two DAGs, thus resulting into the less efficient indexing and loading strategy.

4.2 FHoSP Nesting

After loading and indexing each operand, we can now run the nesting algorithm. Please note that, due to the lack of space, we only describe the algorithm specific to the present paper's use case.

We iterate the graph over each single node appearing in it (Line 6): if the node u is a PHOTO (Line 8) we might extract all the users u associated to that photo and all the possible clusters c_i in which u might fall into. As a result, c_i will be one of the resulting nested graph's nested vertices (Line 14), which will contain u. We can now start to write the nesting index associating cluster c_i to user u and save the same information in primary memory (Line 13). This node visit has the computational complexity of $O(1) + C_p + |in(p)| \cdot C_p$ for each photo p, where C_p is the number of the clusters associated to the photo p; indexing does not provide changes in the computational complexity. We denote such computational complexity as $\mathcal{I}$.

If the node u is a USER for which we have not visited all associated PHOTOs for her/him (Line 20 and 28) or one of her/his followers, then we postpone the analysis once we'll have all the associated cluster information (Line 28 and 35); otherwise, we establish a new nested edge e between the two followers' cluster (Line 31) and nest the original follow's edge inside it (Line 32). We outline two completely opposite scenarios, one **a)** providing the worst case scenario when traversing a non-sorted graph and the other **b)** traversing a topologically sorted graph. **A)** If we visit the peripheral users before the hub nodes in the community and, for each peripheral user, we visit first all the follow edges and then the hasPhoto ones. In this case the computational complexity for each user u is:

$\Sigma_{v \in out_{\texttt{follows}}(u)} |out_{\texttt{hasPhoto}}(v)| + |out_{\texttt{hasPhoto}}(u)| + |out_{\texttt{follows}}(u)|$

and the keys of usersToVisit are the size of the non-peripheral nodes H. **B)** If all the photos are visited before the users, then the

Algorithm 1 Five HOp Separated Patterns Algorithm (FHoSP)

```
1: procedure $\eta5(g)$
2:   visited := $\emptyset$                                                  ▷ Bitmap
3:   usersToVisit := []                                  ▷ HashMap int ↦ unosorted set
4:   clUsersMap := []                                    ▷ HashMap int ↦ unosorted set
5:   NestingIdx := $open(File)$; NestedGraph := $(\mathcal{V}, \mathcal{E})$
6:   for each vertex $u \in V_g$ do
7:     switch $u$.labels do                                            ▷ via $u$.hash
8:       case {PHOTO}:
9:         visited.add($u$)                                                  ▷ $O(1)$
10:        clusters:= { $v \mid (u,v) \in out(u) \land v$.labels={CLUSTERID} }   ▷ $C_u := |out(u)|$
11:        for each edge $(v,u) \in in(u)$ s.t. $v$.labels={USER} do             ▷ $|in(u)|$
12:          for each $c_i$ ∈clusters do
13:            clUsersMap[$c_i$].add($v$); NestingIdx.write($\langle c_i, v\rangle$)
14:            $\mathcal{V}$.add($c_i$)
15:      case {USER}:
16:        skip:=false; friends:= $\emptyset$
17:        for each edge $(u,v) \in out(u)$ with id $e'$ do                    ▷ $|out(u)|$
18:          switch $e'$.labels do
19:            case {hasPhoto}:
20:              if $v \notin$visited then skip:=true                           ▷ $O(1)$
21:            case {follows}:
22:              if not skip then
23:                noSkip:=true
24:                for each edge $(v,w) \in out(v)$ with id $e$ do   ▷ $|out(v)|, v \in out(u)$
25:                  if $e$.labels={hasPhoto} then
26:                    if $w \notin$visited then                                ▷ $O(1)$
27:                      noSkip:=false
28:                      usersToVisit[$v$].add($w, e$)                          ▷ $O(1)$
29:                if noSkip then
30:                  for $(c_i, c_j) \in$ clUsersMap[$u, v$] do[a]              ▷ $C_u \cdot C_v$
31:                    $e := c_i \oplus c_j$; $\mathcal{E}$.add($e = (c_i, c_j)$)
32:                    NestingIdx.write($\langle e, e'\rangle$)
              friends.add($v, e'$)
33:        if skip then
34:          for $(v,e) \in$ friends do                                      ▷ $|out(u)|$
35:            usersToVisit[$v$].add($u, e$)                                  ▷ $O(1)$
36:  for each $v$ ∈key(usersToVisit) do
37:    for $(u, e') \in$ usersToVisit[$v$] do                         ▷ $|in_{\texttt{follows}}(v)|$
38:      for $(c_i, c_j) \in$ clToUserMap[$u, v$] do                            ▷ $C_u \cdot C_v$
39:        $e := c_i \oplus c_j$; $\mathcal{E}$.add($e = (c_i, c_j)$)
40:        NestingIdx.write($\langle e, e'\rangle$)
   return $\langle$NestedGraph,NestingIdx$\rangle$
```

[a] clUsersMap[u,v] is just a shorthand for clUsersMap[u]×clUsersMap[v]

computational complexity for each user u is:

$$\left(\Sigma_{v \in out_{\texttt{follows}}(u)} |out_{\texttt{hasPhoto}}(v)| + C_u \cdot C_v\right) + |out_{\texttt{hasPhoto}}(u)|$$

where C_u is the number of the clusters to which user u is associated via the photos and no element is inserted in the map usersToVisit. Let us denote b_u (b_f) the user to follower (photos) branching factor and k by the average cluster size: **a)** approximates to $\mathcal{U}(b_u b_f + b_u + b_f)$ and **b)** to $\mathcal{U}(b_u b_f + b_f + b_u k^2)$.

The postponed creation of the remaining nested edges is provided at the end of the vertex iteration, once that all the graph data information is collected (Line 39). In the worst case scenario, that is when the majority of the node creation is postponed, then the computational complexity is $\sum_{v \in \text{key}(\texttt{usersToVisit})} C_v \cdot \sum_{u \in in_{\texttt{follows}(v)}} C_u$. Using the shorthands introduced in the former paragraphs and considering that this scenario is triggered for **a)** and never for **b)**, this reduces to $k^2 b_u H$. Last, the nested graph is serialised in secondary memory: this part is omitted in the pseudo-code. The computational complexity of this part is linear with respect to the size of the output nested graph $\mathcal{O}$.

We can finally ask ourselves when the topological sort appears to be the best solution for traversing the graph: if we ignore the cost of $\mathcal{I} + \mathcal{O} + \mathcal{U}(b_u b_f + b_f)$ which is shared among the two scenarios, then the question reduces to ask when $k^2 b_u \mathcal{U} < \mathcal{U} b_u + k^2 b_u H$.

Table 1: Providing the single operand sizes (left) and each graph g used for the benchmark (right).

Sampled	# Vertices	# Edges
Layer1	37	46
Layer2	90	126
Layer3	88	130
Layer4	32	30

Operands	# Vertices	# Edges
Layer1	37	46
Layer1+4	60	76
Layer1+2+4	127	202
Layer1+2+3+4	199	332

We observe that this happens when we are able to guarantee an almost perfect clustering where clusters contain in average at most $\sqrt{\frac{\mathcal{U}}{\mathcal{U}-H}}$ users.

5 EXPERIMENTAL RESULTS

Our preliminary experiments for big data graphs (up to 100 million nodes) show that our approach outperforms the graph nesting implementation over PostgreSQL by at most one order of magnitude, which has already showed to be the best competitor in our previous graph nesting implementation [7] with which we share the same experiment assumptions; Virtuoso and Neo4J provided overall a worse performance than PostgreSQL. Even in this case, we consider the time to *(i)* serialize our data structure (Loading) and (ii) evaluate the query plan (Indexing and FHoSP). For our evaluations, we generate each layer by randomly sampling the Friendster social network graph [20] for 10 users using different seeds, and enriching that with the post (i.e., photo) and tag (i.e., cluster) distribution provided by the LDBC Benchmark [9] and implemented in [2]. Given that this dataset were very small, topological distribution changes did not significantly affected the computation time of the algorithm, which was dominated by the vertices size. We refer to [7] for some nesting examples where the previous THoSP algorithm was performed on both real and synthetic data with different distributions. We kept the analysis to small social networks given that our previous work on user profiling only focused on 10 different devices [16]. For the first and the fourth layer there are many clusters as users, while for the two remaining layers, we assumed that each user might use different devices, and therefore might be part of different photo clusters; the resulting graph layers[2] are described in Table 1 (Sampled). Similarly, we represent the multi-layered graph as one single edge table in PostgreSQL. Therefore, we provided a combination of four possible combination of the layered networks (Table 1, Operands), on top of which we run the FHoSP algorithm and the SQL queries.

Table 2 provides a comparison between FHoSP over the three loading and indexing strategies and PostgreSQL: for our competitor, indexing happens during the query evaluation, and therefore we compare the sum of our indexing and nesting time to PostgreSQL's query evaluation. Please note that the loading time in PostgreSQL corresponds to loading the edges' tables for all the layers within one single table. We observe that, for small datasets, the loading time using the the topology ordering (τ) is nearly comparable to the cost of loading the graph without ordering the graph in primary memory (ι) or ordering the vertices by hash value ($\hbar$). Nevertheless, FHoSP loading time is comparable to PostgreSQL's. Finally, we might observe that a specific indexing strategy does not provide

[2] The dataset is available at https://rebrand.ly/fhospdata.

Table 2: Separating Loading, Indexing and Nesting time (ms) for the nesting algorithm for the present use case.

Loading	**FHoSP** (+ι)	**FHoSP** (+$\hbar$)	**FHoSP** (+τ)	**PostgreSQL**
Layer1	**0.228**	0.386	0.231	3.133
Layer1+4	0.677	0.334	**0.322**	3.323
Layer1+2+4	**0.686**	0.810	0.705	3.833
Layer1+2+3+4	1.951	**1.145**	1.208	4.013

Indexing+Nesting	**FHoSP** (+ι)	**FHoSP** (+$\hbar$)	**FHoSP** (+τ)	**PostgreSQL**
Layer1	**0.291**	0.394	0.343	8.337
Layer1+4	**0.415**	0.440	0.485	8.557
Layer1+2+4	1.274	1.305	**1.153**	10.218
Layer1+2+3+4	1.944	1.736	**1.695**	20.125

significant advantages for small datasets even though all the proposed solutions outperform PostgreSQL's associated query plan by one order of magnitude. Further tests should be also carried out on bigger datasets in order to strongly substantiate this preliminary experimental evidence.

6 DISCUSSION

In this paper we did not discussed and evaluate the quality of the final nesting outcome: we already discussed the quality of the clustering approach in [16], while the present algorithm takes an output from the previous phase. Therefore, the present paper will only focus on the computational complexity of the multi-network nesting. Our previous results in [16] suggested that the sensor pattern noise is a reliable characteristic to solve user profiles resolution problem. As a result, we present a paper providing a preliminary analysis comparing a nested graph ad-hoc implementation to current SQL query plans (i.e., PostgreSQL's). Albeit the dataset of choice was small, the preliminary study conducted in the present paper suggests that the proposed approach is promising: while our serialization algorithm is comparable to PostgreSQL's, our proposed algorithm (FHoSP) is always ten times faster than PostgreSQL's query plan. To simplify the current problem, we assumed that all the OSNs had the same schema and that the relationships among users, photos, and cluster_id are always both labelled as the same and similarly represented across multiple different OSNs. On the other hand, different OSN may provide the same information using a different representation, thus resulting in a different schema representation. In this case, we might use the Ϙ operator [5] for nesting multiple graphs after aligning different OSN's schemas as suggested in the current literature.

7 CONCLUSIONS

Social networks have always fascinated researchers who are interested in various problems, like information dissemination, missing data and visualisation. Recently, there has been a growing interest in multi-layer networks, where the individual networks are mutually connected. In this paper, we present a preliminary study on approximate nesting of multiple OSN graphs based using the sensor pattern noise of the shared images. The clustering approach classifies the users according to the different smartphones used to capture and share the images, resulting in multiple *inter-layer edges* across layers. In our future work, we will also take into account multiple possible user descriptors to target user profiles resolution, and the possibility that the clustering phase is directly integrated into the graph nesting query plan. As a result, we would need to introduce approximate graph matching for covering multiple descriptions and generalise the currently provided nesting algorithm for any given graph nesting task expressible by Ϙ. The results are promising and suggest that further research should be conducted to evaluate our approach over bigger datasets over real data.

ACKNOWLEDGMENT

This work was partially supported by the H2020 framework project OPERANDUM [grant number 776848].

REFERENCES

[1] N. Al Mutawa, I. Baggili, and A. Marrington. Forensic analysis of social networking applications on mobile devices. *Digital Investigation*, 9:S24–S33, 2012.

[2] G. Bagan, A. Bonifati, R. Ciucanu, G. H. L. Fletcher, A. Lemay, and N. Advokaat. gMark: Schema-driven generation of graphs and queries. *EE Trans. Knowl. Data Eng*, 29(4):856–869, 2017.

[3] G. Baldini and G. Steri. A survey of techniques for the identification of mobile phones using the physical fingerprints of the built-in components. *IEEE Communications Surveys & Tutorials*, 19(3):1761–1789, 2017.

[4] S. Bartunov, A. Korshunov, S.-T. Park, W. Ryu, and H. Lee. Joint link-attribute user identity resolution in online social networks. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, Workshop on Social Network Mining and Analysis. ACM*, 2012.

[5] G. Bergami. *A new Nested Graph Model for Data Integration.* PhD thesis, Alma Mater Studiorum – University of Bologna, 2018.

[6] G. Bergami, M. Magnani, and D. Montesi. A join operator for property graphs. In *EDBT/ICDT Workshops*, 2017.

[7] G. Bergami, A. Petermann, and D. Montesi. THoSP: an algorithm for nesting property graphs. In *Proceedings of the 1st ACM SIGMOD Joint International Workshop GRADES/NDA*, page 8. ACM, 2018.

[8] M. E. Dickison, M. Magnani, and L. Rossi. *Multilayer social networks.* Cambridge University Press, 2016.

[9] O. Erling, A. Averbuch, J. Larriba-Pey, H. Chafi, A. Gubichev, A. Prat, M.-D. Pham, and P. Boncz. The ldbc social network benchmark: Interactive workload. In *SIGMOD '15*, pages 619–630, New York, NY, USA, 2015. ACM.

[10] G. Estrada-Rodriguez, E. Estrada, and H. Gimperlein. Metaplex networks: influence of the exo-endo structure of complex systems on diffusion. *CoRR*, arXiv:1812.11615, 2018.

[11] K. A. Kumar and P. Efstathopoulos. Utility-driven graph summarization. volume 12, pages 335–347. VLDB Endowment, 2018.

[12] J. Lukáš, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Trans. Inf. Forensics Security*, 1(2):205–214, 2006.

[13] F. M. Naini, J. Unnikrishnan, P. Thiran, and M. Vetterli. Where you are is who you are: User identification by matching statistics. *IEEE Trans. Inf. Forensics Security*, 11(2):358–372, 2016.

[14] F. Norouzizadeh Dezfouli, A. Dehghantanha, B. Eterovic-Soric, and K.-K. R. Choo. Investigating social networking applications on smartphones detecting facebook, twitter, linkedin and google+ artefacts on android and ios platforms. *Australian journal of forensic sciences*, 48(4):469–488, 2016.

[15] O. Peled, M. Fire, L. Rokach, and Y. Elovici. Entity matching in online social networks. In *2013 International Conference on Social Computing*, pages 339–344. IEEE, 2013.

[16] R. Rouhi, F. Bertini, and D. Montesi. A cluster-based approach of smartphone camera fingerprint for user profiles resolution within social network. In *Proceedings of the 22nd International Database Engineering & Applications Symposium*, pages 287–291. ACM, 2018.

[17] K. Shu, S. Wang, J. Tang, R. Zafarani, and H. Liu. User identity linkage across online social networks: A review. *Acm Sigkdd Explorations Newsletter*, 18(2):5–17, 2017.

[18] S. S. Skiena. *The Algorithm Design Manual.* Springer, 2nd edition, 2008.

[19] P. Slater. Inconsistencies in a schedule of paired comparisons. volume 48, pages 303–312, 1961.

[20] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 745–754, 2012.

Chi Squared Feature Selection over Apache Spark

Mohamed Nassar, Haidar Safa, Alaa Al Mutawa, Ahmed Helal, Iskander Gaba
Computer Science Department
Faculty of Arts and Sciences
American University of Beirut (AUB)
[mn115|hs33|aja35|amh90|img02]@aub.edu.lb

ABSTRACT

We live in the age of big data and distributed computing. The current large scale computation frameworks are based on a scaling-out approach for distributing tasks over a cluster of commodity machines. Apache Spark is one of these frameworks that has excelled in many computational tasks. Implementation of statistical learning algorithms over Spark is a challenging task. A bad implementation may lead to a significant decrease in performance and a waste of cluster time and money. Poor performance is mostly due to a lack of understanding of the data in hand and Spark's underlying mechanisms more than it is due to a deficit in the framework itself. In this paper, we consider the use case of χ^2 feature selection which is very popular in supervised learning pipelines. Our implementation follows the algorithm of the Scikit-learn Python machine learning library which is different than the algorithm used by the Spark machine learning library. The Spark ML library implementation of χ^2 feature selection accepts only categorical features. Our alternative implementation is more suitable for numerical features. We experiment in particular with features of high sparsity such as n-gram counts. We study the best partitioning scheme of the data and the optimal number of partitions. Our experiments are run over the Databricks platform.

CCS CONCEPTS

• **Computing methodologies → Feature selection**; **Distributed computing methodologies**;

KEYWORDS

Feature Selection, Chi-Squared, Apache Spark, Big Data, Cluster Computing, Map-Reduce

ACM Reference Format:
Mohamed Nassar, Haidar Safa, Alaa Al Mutawa, Ahmed Helal, Iskander Gaba. 2019. Chi Squared Feature Selection over Apache Spark. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3331076.3331110

1 INTRODUCTION

We live in the age of big data and distributed computing. The current large scale computation frameworks are based on a scaling-out approach for distributing tasks over a cluster of commodity machines. These machines are relatively slow with cheap interconnects and abundant failures. Scaling-out has gained momentum face to the huge cost associated with scaling-up systems. Map-Reduce is the programming paradigm of scaling-out systems such as Apache Hadoop and Apache Spark. Spark has extended the Hadoop framework by improving performance and adding more utilities such as the ability to work with dataframes in a SQL-like manner. Feature selection is a very important building block in any machine learning pipeline. Its aim is to remove non-informative features and select the ones that are most useful for prediction. Feature selection helps reduce the training time, improve accuracy and avoid overfitting the trained model. Designing the best distributed implementation of feature selection is a challenging task. In this paper, we present an alternative implementation of χ^2 feature selection over Apache Spark, based on the algorithm used in Scikit-learn, and evaluate it over the Databricks platform. χ^2 feature selection is based on hypothesis testing which is a powerful tool in statistics. Hypothesis testing determines whether a result is statistically significant, or in other words, whether it occurred by chance or not.

The Spark machine learning library implements χ^2 feature selection only for categorical data based on building the standard χ^2 contingency table for each feature. In contrast the scikit-learn Python library implementation of χ^2 accepts numerical features such as term counts in document classification. Scikit-learn does not build the complete contingency table. Rather it uses a simplified χ^2 formula to measure the dependence between a feature and the label (or class) as being two stochastic variables. Even though this may not exactly reflect the standard theoretical framework of the algorithm, it is widely considered as very useful in practise. Scikit-learn is however designed to work on a single machine. We propose an efficient and distributed implementation of the same algorithm over Apache Spark. We experiment with datasets of different sizes and sparsity. In particular we study the best partitioning scheme of the data and the optimal number of partitions.

The remaining of this paper is organized as follows. In section 2 we review the Spark framework and data types. χ^2 feature selection is detailed in section 3. We discuss current implementations and present our alternatives in section 4. Experimentation and results are addressed in section 5. Finally section 6 concludes the paper and sheds light on future work.

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06…$15.00
https://doi.org/10.1145/3331076.3331110

2 BACKGROUND ON APACHE SPARK

Apache Spark [15] is a unified engine that makes big data processing tractable through parallel computation, in-memory processing and on the fly optimization. It is a fault-tolerant and general purpose cluster computing system providing APIs in Java, Scala, Python and R. Spark runs workloads on large scale clusters 100 times faster than its Hadoop predecessor. It was originally designed as a more general computing engine to specifically address three limitations in previous map-reduce frameworks: (1) capability to deal with general executing graphs and iterative algorithms that make many passes through the same data, (2) real-time streaming: compute tasks incrementally as new data arrives, and (3) interactive queries which makes it accessible through a notebook-like interface. Spark developers chose to keep a small core while contributing additional features through libraries such as MLlib, Spark SQL, Streaming and GraphX. The data sharing abstraction of Spark is called "Resilient Distributed Dataset", or RDD. RDD is a distributed collection of JVM objects that are strongly typed and support functional operators such as `map`, `filter`, `reduceByKey`, `flatMap`, etc. Pair RDD refers to an RDD of key/value pairs. RDD are however considered unstructured since the internal structure of the RDD objects is unknown. Dataset is a new interface that provides the benefits of RDDs along with the benefits of Spark SQL's optimized execution engine. A Dataframe is a Dataset of rows organized into named columns. A Dataframe is conceptually equivalent to a table in a relational database and supports operations such as `show, select, agg, groupBy, join,` etc. For more details about these three APIs (RDD, Dataset and Dataframe), we refer the reader to [5]. Spark can perform any parallel computation based on its map-reduce general paradigm. The map operation allows local computation tasks by transferring the code to the nodes hosting the data. The reduce operation allows all-to-all communication which can emulate any message exchange, even though sometimes inefficiently. Yet a series of smart optimizations made Spark's performance comparable to many specialized distributed computation systems. For instance, Spark achieves fault tolerance efficiently by using lineage graphs and avoid storing data that can be-recomputed. Lost partitions are re-computed from lineage graphs in case of a failure.

Programming under Spark might be seen as less flexible than under other distributed computation frameworks such as message passing ones. Still, some optimization techniques are available. For instance the developer can customize the partitioning of the data among the nodes. A smart partitioning may bring substantial performance gains in face of shuffles. Shuffling is moving data from one node to another to be grouped with its key. Shuffling is required by some operations such as `reduce` or `groupByKey`. By default, Spark uses hash partitioning which attempts to spread the data evenly across partitions based on the key hash. Other options to control the partitioning of pair RDDs are to set the number of partitions, use a range partitioner or create a new customized partition scheme.

3 χ^2 FEATURE SELECTION

The χ^2 statistics [14] are commonly used to rank binary, discrete and nominal features. The score of a feature is a measure of how much the expected count E and the observed count O deviate from each other. The expected sum assumes the independence of any of the categories of the feature f from the class label c:

$$E(f = v_i, c) = S(f = v_i)\frac{\text{count}(c)}{\text{total\_count}} = S(f = v_i)P(c)$$

where: $S(f = v_i)$ is the number of occurrences of the category v_i among all the data instances, count(c) is the count of data instances belonging to the class c, total_count is the total number of instances in the dataset, and $P(c)$ is the likelihood that a randomly drawn data instance would belong to c.

The observed sum $O(f = v_i, c)$ is simply the count of v_i occurrences for all the data instance belonging to c in the dataset. Assuming that the dataset has n data instance and k different classes, the overall χ^2 statistic for a feature f with m different categories is given by:

$$\chi^2(f) = \sum_{c=1}^{k}\sum_{i=1}^{m}\frac{(O(f = v_i, c) - E(f = v_i, c))^2}{E(f = v_i, c)}$$

The same formula is used by the Spark ML implementation [4].

$\chi^2(f)$ is used to test the independence of f and the class label c. The confidence of rejecting the independence hypothesis gets higher for higher values of $\chi^2(f)$, indicating that the feature f is very likely to be correlated with the class label c. Therefore a straightforward selection procedure is to sort the features in descending order of their χ^2 statistics and the top ones are selected.

Note that the calculation of the χ^2 test involves only arithmetic operations such as addition and multiplication. For a dataset of N instances (or rows), k classes and n features of m categories each, the complexity of computing the χ^2 statistics is $O(n * N)$ time and $O(n*m*k)$ space. Using this formulation for numerical features will be very inefficient in terms of computation time since all different values of the feature are considered different categories (n would be large). Sometimes binning is used to decrease the number of categories. For example, in case of n-grams, it is common to work with $4 * 10^6$ features. If each feature has 100 categories (or bins) and we have 10 classes, the needed memory is proportional to $4 * 10^9$. In fact, the current Spark ML implementation splits features into groups of 1000 features and treats them sequentially, making needed space proportional to $4 * 10^6$ only in this scenario. Therefore the contingency tables can be stored on a single machine for each iteration. Still for $N = 10^3$ data points the computation time is proportional to $4 * 10^9$. Our experience is that this scenario would be exhausting for a small Spark cluster with the current Spark ML implementation. It takes long hours.

The implementation of scikit-learn [3] does not build a complete contingency table per feature. The features are numerical and their values represent the number of occurrences of the feature in each data instance, for example the number of appearances of a term in a document. The following formula is used:

$$\chi^2(f) = \sum_{c=1}^{k}\frac{(O(f, c) - E(f, c))^2}{E(f, c)}$$

where $O(f, c)$ is the observed sum of numerical value of f across all data instances. $E(f, c)$ is the expected sum assuming independence in between the feature and the class label. For a dataset of N instances (or rows), k classes and n features, the complexity of computing the χ^2 statistics is $O(n * N)$ time and $O(n * k)$ space. In case of the aforementioned n-grams example, the computation time is proportional to $4 * 10^9$. However, exploiting sparsity in the data makes it much lesser. Our Spark implementation is able to perform it in a few minutes as will be shown in the experiments section.

To illustrate the difference between the two approaches, let's consider the case of a binary feature with the following contingency matrix:

		class	
		c_0	c_1
category	f	$A = 3$	$B = 5$
	$\neg f$	$C = 2$	$D = 10$

A, B, C and D are the observed count of respectively (f, c_0), (f, c_1), $(\neg f, c_0)$ and $(\neg f, c_1)$. The size of the dataset is $N = A + B + C + D$. We consider the numerical example where $A = 3,\ B = 5,\ C = 2,\ D = 10,\ N = 20$. The two libraries will compute the χ^2 scores as shown in the following table:

	Spark ML[4]	**Scikit-Learn**[3]
formula	$\frac{n(AD-BC)^2}{(A+C)(A+B)(C+D)(B+D)}$	$\frac{(AD-BC)^2}{(A+B)(A+C)(B+D)}$
score	10/9	2/3

The Python API code to generate the scikit-learn score is shown next:

```
from sklearn.feature_selection import chi2
import numpy as np
X = np.array([1]*8 + [0]*12)
X = X.reshape(-1, 1)
Y = [0]*3 + [1]*5 + [0]*2 + [1]*10
chi2_score, p_value = chi2(X,Y)
```

The Scala API code to generate the Spark ML score is shown next:

```
import org.apache.spark.ml.linalg.Vectors
import org.apache.spark.ml.stat.ChiSquareTest
val labels = Seq(0, 0, 0, 1, 1, 1, 1, 1,
    0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
val feature =
(for (i <- 0 until 8 ) yield Vectors.dense(1))
++
(for (i <- 0 until 12 ) yield Vectors.dense(0))
val data = labels zip feature
val df = data.toDF("label", "features")
ChiSquareTest.test(df, "features", "label")
    .show
```

4 IMPLEMENTATION

In this section we explore, in a more technical way, the different implementations of the computation of χ^2 statistics and we propose our own alternative.

4.1 Scikit-learn: Implementation for a single machine

Scikit-learn is a machine learning Python library. The scitkit-learn implementation starts by building the class-instance contingency matrix. An element e_{ij} of this matrix is equal to 1 if instance j belongs to class i, 0 otherwise. It proceeds with the multiplication of two matrices: the class-instance contingency matrix of size $k * N$ and the instance-feature contingency matrix $N * n$. The result is the class-feature contingency matrix of size $k * n$. The class-feature matrix contains the observed sums for all class-feature pairs. The matrix multiplication is very efficient in the case where the matrices fit in memory. Scikit's reliance on underlying numerical libraries, namely `numpy` and `scipy`, makes this matrix multiplication, and hence the whole procedure very fast.

4.2 Spark ML/MLlib implementation: Categorical features

Spark ML supports Pearson's Chi-squared (χ^2) tests for independence. The API `ChiSquareTest` takes for input a dataframe of categorical labels and categorical features. The ML implementation is just a wrapper that transforms the dataframe into an RDD of `LabeledPoint` and passes it to the old implementation (`ChiSqTest`) of the Mllib library. The `ChiSqTest` implementation is marked as experimental and belongs to the `org.apache.spark.mllib.stat.test` package. The RDD is passed to the `chiSquaredFeatures` function. The function starts by counting the number of features and building an array of type `ChiSqTestResult` to store the results for each feature (referred to as `col` in the implementation). The maximum number of allowed categories per feature (and the maximum number of allowed distinct labels) is fixed at 10,000. The function groups the features into batches of 1000 features each and processes them sequentially. The transformation `mapPartitions` is followed by the action `countByValue` in order to generate the (category, label) `pairCounts`. For each feature, the `pairCounts` are accumulated in a contingency matrix. The Breeze linear algebra library is locally used to compute the final χ^2 results.

4.3 Our implementation over Spark: Numerical features

We propose a Scala implementation of χ^2 feature selection for Spark. Our implementation is different than the Spark ML library in the following points: (1) It is based on the scikit-learn formulation for feature selection. This choice allows some optimizations which are not possible in the Spark ML implementation, namely taking sparsity of features in consideration. (2) It is mainly focused on numerical features. (3) It uses the capabilities of the dataframe API whenever it is possible. On the other side, the Spark ML implementation is completely based on the RDD capabilities. (4) The implementation is completely distributed without recurring to the Breeze library. Breeze would require that the categories vectors and the contingency matrix are available locally and can fit within a single node. We do not make such assumptions.

Our implementation has actually two different versions: the dense version and the sparse version. The dense version takes each feature vector as a dense vector and does not assume sparsity. By

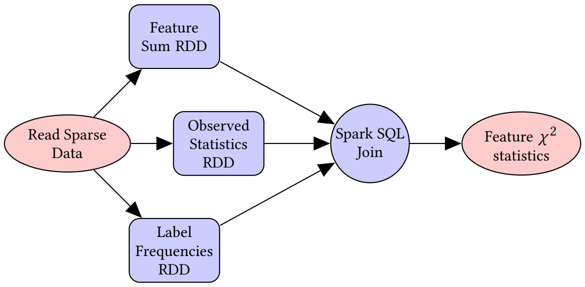


Figure 1: Flow-chart of the sparse version in Scala/Spark

Table 1: Summary of Datasets

Dataset	Size	Rows (N)	Features (n)
1	4.5 MB	32,561	288,196
2	82.3 MB	748,401	1,163,024
3	1.3 GB	4,203,876	11,557,504
4	2.67 GB	8,407,752	20,216,830
5	24.9 GB	45,840,617	999,999

sparsity we mean that the value of a feature could be 0 in a large number of rows (or samples). We measure it as the ratio of zero entries to non-zero entries in the instance-feature matrix. This is often the case in term counts and n-gram features. The sparse version takes benefit of this property to speed up the computation. Note that sparsity can also be exploited in the case of categorical features. 0 would be considered as a distinct category but the number of 0's occurrences would be computed as N minus the sum of the occurrences of all the other categories.

The implementation flow chart is shown in Figure 1. It starts by loading the data in sparse format (LibSVM format), then it builds three dataframes (or tables): (1) the first dataframe has the total sum for each feature, (2) the second dataframe has the observed pair counts for each (label, feature) tuple, and (3) the third dataframe has the frequency of appearance of each label in the dataset. RDD capabilities are used to generate these dataframes. The last step is to build the contingency table out of the three dataframes using two joins and a user defined function `chi2UDF`. Dataframe capabilities are used at this stage.

5 EXPERIMENTS

We have experimented with datasets of different propoerties such as size, number of samples, number of features and sparsity [1]. The number of classes is $k = 2$ for all the datasets. The datasets properties are listed in Table 1. We store the datasets in an AWS S3 bucket to be easily retrieved by the Spark Databricks cluster. All the datasets are in LibSVM sparse format. The LibSVM format is efficient for storing sparse datasets since only non-zero values are stored along with their indices. Each row represents a data instance (or sample) and has the following form:

```
<label> <index1>:<value1> <index2>:<value2> ...
```

For example The line '`+1 2:1 4:2`' represents a data instance with class +1 and feature vector [0, 1, 0, 2].

We have used the databricks platform for our experiments. Databricks provides a tiny cluster for preliminary experiments known as the community edition. The paid edition provides a pre-configured Spark cluster based on AWS machines. The specifications of the two options that we have used are shown in Table 2.

Table 2: Summary of Cluster Specifications

Community Edition (CE)	Paid Edition (PE)
0 workers	2-8 workers 128.0-512.0 GB Memory 16-64 Cores 3.6-14.4 DBU
1 Driver: 6.0 GB Memory 0.88 Cores 1 DBU Free	1 Driver: 64.0 GB Memory 8 Cores 1.8 DBU Cost = $0.40 per DBU
Scala 2.11 Spark 2.3.1 Databricks Runtime 4.3	

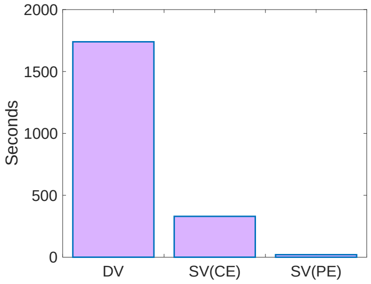


Figure 2: Runtime for dataset #3 with Dense Version *DV*, Sparse Version – Community Edition *SV(CE)*, and Sparse Version – Paid Edition *SV(PE)*

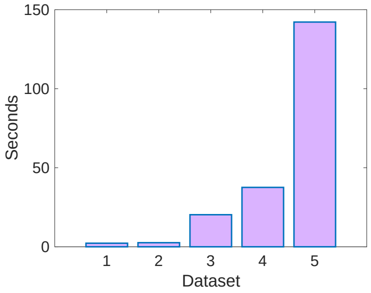


Figure 3: Runtime for the 5 datasets over the PE cluster

We compare the performance of our two versions (DV and SV) over the two clusters (CE and PE) based on a common dataset. The results for dataset #3 are shown in Figure 2. It is clear that taking benefit of sparsity radically improves performance. The processing takes only a few minutes over the real cluster (PE). The runtime for the different datasets over the PE cluster is shown in Figure 3.

We have also experimented with different partitioning schemes. For Dataset #5 we have set the number of partitions to 80, 160, 200, or the default settings. The default settings are defined by Spark and depends on the dataset size and the number of available cores [2]. We plot the runtime for different size percentages (20% to 100%) of the Dataset #5 (~25GB) in Figure

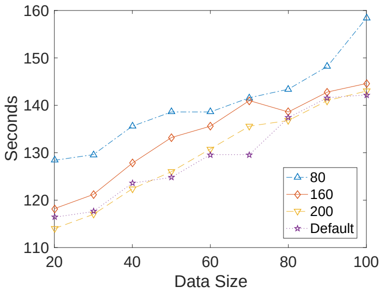


Figure 4: Effect of Partitioning - dataset #5

4. Results show that increasing the number of partitions helps decreasing the overall runtime. The runtime increases softly with respect to the data size. This is a good indication that the distributed implementation over Spark is working well. We have also experimented with re-partitioning the data based on the class as a key. The re-partitioning has shown similar performance in the case of our datasets. We plan to investigate partitioning in more depth in future work.

6 RELATED WORK

There is an increasing interest in implementing distributed linear algebra and machine learning routines and data structures over Hadoop [13] and Apache Spark [7]. For nice lessons learned while implementing a basic machine learning algorithm we refer the reader to the talk in [8]. Another area of interest focuses on automatic building and optimization of complex Spark applications. In [12], a component-based framework for composing independently developed Spark applications is proposed. This framework is equipped with a transformation-based optimizer that takes a Spark program and generates a state-space of semantically equivalent programs by applying a set of rewrite rules. The best semantic-equivalent program is returned based on a set of pre-selected strategies.

Selecting the maximum quality levels to execute given Spark applications with quality of service constraints is investigated in [10]. ASC [16] is an automatic checkpoint algorithm that optimizes the selection, frequency and timing of RDD persisting in a long lineage. A checkpoint cuts off the lineage and save the data which is required in the incoming computations. The solution is shown to have a small overhead with respect to the performance benefits it brings in case of failures. Spark SQL is based on a highly extensible optimizer so-called Catalyst [6]. It is built using features of the Scala programming language, that makes it easy to add composable rules, and control code generation. Catalyst is used to build a variety of features tailored for the complex needs of modern data analysis. VEGA [9] is an Apache Spark framework for optimizing a series of similar Spark programs. These programs are likely originated from an exploratory data analysis session. Data scientists can leverage Vega to significantly reduce the amount of time when modifying and re-executing Spark programs over large datasets. HYLAS [11] is a tool for automatically optimising Spark queries embedded in source code via the application of semantics-preserving transformations. Hylas can identify certain computationally expensive operations and transform them to better alternatives, which leads to signification improvements in execution time. The contribution of this paper is different since it considers a very specific task and contrasts it to the standard library implementation in Spark ML.

7 CONCLUSION AND FUTURE WORK

In machine learning, feature engineering is very important to reduce the training time, improve accuracy, and avoid overfitting the training data. While most practitioners run their machine learning algorithms on specialized GPU farms rather that on cluster computing platforms such as Hadoop and Spark, cluster computing is nevertheless essential for data cleaning, pre-processing and feature selection. One of the popular statistical models to select the most relevant features is the Chi-Squared test. In this paper we compared the implementations of Chi-Squared feature selection in scikit-learn and Spark ML. We proposed a new implementation for numerical features over Apache Spark. We showed the performance of our approach using different real-world data sets. We also studied the effects of sparsity and partitioning. The benchmarking is performed over the Databricks cloud computing platform. It is worth noting that our approach runs much faster than the current Spark ML one for three main reasons: (1) we use a formulation with lower algorithmic complexity, (2) we do not have any sort of sequential processing of features in chunks like in the Spark ML implementation, and (3) we use the optimized dataframe API and Spark SQL routines whenever possible. In future work, we aim to study and implement advanced linear algebra, feature selection and machine learning algorithms over Apache Spark.

ACKNOWLEDGMENT

This work was supported partially by a grant from the university research board of the American university of Beirut (URB-AUB-2018/2019).

REFERENCES

[1] Libsvm data: Classification, regression, and multi-label. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.
[2] Partitioning in apache spark. https://medium.com/parrot-prediction/partitioning-in-apache-spark-8134ad840b0.
[3] sklearn.feature_selection.chi2. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html. Accessed: 2019-03-11.
[4] Spark docs - basic statistics - hypothesis testing. https://spark.apache.org/docs/2.2.0/ml-statistics.html. Accessed: 2019-03-11.
[5] A tale of three apache spark apis: Rdds vs dataframes and datasets. https://databricks.com/blog/2016/07/14/a-tale-of-three-apache-spark-apis-rdds-dataframes-and-datasets.html.
[6] M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, et al. Spark sql: Relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1383–1394. ACM, 2015.
[7] R. Bosagh Zadeh, X. Meng, A. Ulanov, B. Yavuz, L. Pu, S. Venkataraman, E. Sparks, A. Staple, and M. Zaharia. Matrix computations and optimization in apache spark. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 31–38. ACM, 2016.
[8] L. Dali. Lessons learned while implementing a sparse logistic regression algorithm in apache spark. https://databricks.com/session/lessons-learned-while-implementing-a-sparse-logistic-regression-algorithm-in-apache-spark.
[9] M. Interlandi, S. D. Tetali, M. A. Gulzar, J. Noor, T. Condie, M. Kim, and T. Millstein. Optimizing interactive development of data-intensive applications. In *Proceedings of the Seventh ACM Symposium on Cloud Computing*, pages 510–522. ACM, 2016.
[10] M. Jaber, M. Nassar, W. A. R. Al Orabi, B. A. Farraj, M. O. Kayali, and C. Helwe. Reconfigurable and adaptive spark applications. In *CLOSER*, pages 84–91, 2017.
[11] Z. A. Kocsis, J. H. Drake, D. Carson, and J. Swan. Automatic improvement of apache spark queries using semantics-preserving program reduction. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 1141–1146. ACM, 2016.
[12] Z. Shmeis and M. Jaber. Fine and coarse grained composition and adaptation of spark applications. *Future Generation Computer Systems*, 86:629–640, 2018.
[13] M. Wang, S. B. Handurukande, and M. Nassar. Rpig: A scalable framework for machine learning and advanced statistical functionalities. In *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, pages 293–300. IEEE, 2012.
[14] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Icml*, volume 97, page 35, 1997.
[15] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, et al. Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11):56–65, 2016.
[16] W. Zhu, H. Chen, and F. Hu. Asc: Improving spark driver performance with automatic spark checkpoint. In *2016 18th International Conference on Advanced Communication Technology (ICACT)*, pages 607–611. IEEE, 2016.

Performance evaluation on encrypted – non encrypted database fields containing IoT data

Christodoulos Asiminidis†
Laboratory Team of Distributed Microcomputer Systems,
Dept. of Mathematics,
University of Ioannina
chrisasimi@gmail.com

Ioannis Georgiadis
Laboratory Team of Distributed Microcomputer Systems,
Dept. of Mathematics,
University of Ioannina,
johngeo235@gmail.com

Dimitrios Syndoukas
Dept. of Business Administration
Techn. Edu. Inst. of Western Macedonia, Grevena, Greece
dsyn@teiwm.gr

George Kokkonis
Dept. of Business Administration
Techn. Edu. Inst. of Western Macedonia, Grevena, Greece
gkokkonis@teiwm.gr

Sotirios Kontogiannis†
Laboratory Team of Distributed Microcomputer Systems,
Dept. of Mathematics,
University of Ioannina
skontog@cc.uoi.gr

ABSTRACT

Encrypted data transmitted from secure wireless channels as part of Internet of Things (IoT) infrastructures are mainly stored in database tables using plaintext records. As IoT data content confidence in supportive A.I. actions radically increases, the selection and use of database encryption is essential in terms of data integrity safety as well as IoT operations or decisions validity and non repudiation. Taking into account the amount of stored IoT data, data filtering and processing overhead needed for aptly decision making, it is essential for a developer to seriously take into account encryption-decryption queries overheads for big IoT data processing tasks, as well as extra time delays on encryption-decryption operations, since these translate into idle or receiving states energy expenditures at the wireless IoT motes.

KEYWORDS

Database encryption, Big data, IoT, performance evaluation

CCS CONCEPTS

Security and privacy → Database and storage security → Management and querying of encrypted data

IDEAS'19, June 10 -12, 2019, Athens, Greece
© 2019 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-6249-8/19/06.
https://doi.org/10.1145/3331076.3331097

1 Introduction

IoT is an ongoing technological tendency. It is estimated that there will be more than 9 billion devices on the 2020, most of which will operate autonomously based on AI algorithms decisions fed by IoT data. Up to now the decision logic of IoT actuators and controllers is performed by the Application services or cloud services layers, but this is about to change. Since contemporary IoT devices turn away from the 8, 16bit microcontrollers to low power 32, 64bit microprocessor devices of big flash storage and high throughput transponders, only the data sets to be left to the service end while the algorithmically logic gradually migrates to the motes end.

2 Performance evaluation

The performance evaluation scenarios include three types of data encryption/decryption stress tests performed to a PostgreSQL database server. The database system is a Dual-core 3.2GHZ system with 4GB RAM and 500GB storage size, of similar characteristics to the Microsoft Azure A2 instance provided as a business solution for small to medium databases [3]. Since commonly used Database benchmarks [4, 5, 6] do not include encrypted columns database tests, authors created a set of Python scripts to test PostgreSQL AES-128bit encryption currently used by IoT wireless nodes data transmissions. Three different cases are examined into a table of encrypted AES columns following a per field relational schema and a table of an AES column that follows the JSONB schema less form and includes all IoT measurements using the JSON notation [1, 2]

Table 1: Average insertion, selection and aggregation for relational and JSOB fields on encrypted and non-encrypted data

1M records 5M records	Relational plaintext	JSONB plaintext	Relational encrypted	JSONB encrypted
Average Insertion Time (ms)	4.12 3.81	9.5 8.25	8.85 8.64	9.69 10.55
Average Selection Time (ms)	3424.72 3595.73	8818.40 34090.18	9333.66 59356.59	7095.92 37378.66
Average Aggregation Time (ms)	245.18 2652.28	646.19 937.47	10005.12 49846.93	9722.44 45896.58

The comparison results between data inserted into relational and JSONB on encrypted and non-encrypted fields have shown that for 1M records, the relational processing time is the fastest one to 4.12988 ms. The JSONB encrypted processing time underperformed by giving average processing time to 9.69483 ms. For 5M records, relational plaintext remains the fastest amongst the rest and specifically, the average execution time is to 3.81368 ms. In contrast, the JSONB encrypted processing time gave the worst results to 10.55648 ms. The select experimental results have shown that for 1M records relational plaintext is the faster one to 3424.7229 ms and the relational encrypted is the slowest one to 9333.6668 ms. For 5M records, the relational processing time on plaintext gives the best performance to 3595.73602 ms and the relational encrypted the worst one to 59356.59885 ms. The aggregation experimental scenario results for 1M records have shown that the relational processing time on plaintext performed the best and to 245.18394 ms in contract with the relational execution time on encrypted data which increased to 10005.12385 ms. For 5M records, results showed that in the case of aggregation the JSONB processing time on plaintext performed the best to 937.47463 ms and the worst case amongst the four is the relational processing time on encrypted data which increased to 49846.93503 ms.

3 Conclusions

Comparisons between relational tables of plaintext IoT data and relational tables of AES-128 encrypted records have shown that the mean processing time for IoT data inserts increases 2-2.5 times. For single column decrypt-select queries the mean processing time increases 2%-92% of the corresponding plaintext queries processing time for 1M and 5M records accordingly. The authors also pinpoint an additive factor of increase of the select queries processing time, proportional to the number of the decrypted fields per query (additive factor of 1.5x, where x=number of decrypted fields). A single column aggregation function performed on an encrypted table field cost 25-18.7 times more processing time than on its corresponding plaintext field, for 1M and 5M records accordingly. Comparisons between JSONB tables of plaintext fields and JSONB tables of AES-128 encrypted fields have shown that the mean processing time for IoT data inserts remains the same. For single column decrypt-select queries the mean processing time increases 8% the corresponding plaintext queries processing time for 1M and 5M records accordingly. Finally, cross-comparisons between JSONB tables and relational tables of encrypted AES-128 IoT data have shown that for data inserts the mean processing time for JSONB tables is 25% more than the relational tables for 1M and 5M records accordingly. For IoT data selects, JSONB processing time is 40-60% times more than the relational table select queries on a single field for 1M and 5M records accordingly.

ACKNOWLEDGEMENTS

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T1EDK-02374).

ACM Reference Format

Christodoulos Asiminidis, Ioannis Georgiadis, Dimitrios Syndoukas, George Kokkonis and Sotirios Kontogiannis, 2019. Performance evaluation on encrypted – non encrypted database fields containing IoT data.*In proceedings of IDEAS ’19, Athens, Greece, June 10-12, 2019 (IDEAS ‘19),* 2 pages. https://doi.org/10.1145/3331076.3331097

REFERENCES

[1] Ch. Asiminidis, G. Kokkonis, S. Kontogiannis. (2018) “Database Systems Performance Evaluation for IoT Applications”, International Journal of Database Management Systems -IJDMS, 10(6)

[2] Ch. Asiminidis, G.Kokkonis, S. Kontogiannis, (2019) "Managing IoT data using relational schema and JSON fields, a comparative study", IOSR Journal of Computer Engineering (JCE), 20(3)

[3] Microsoft Azure SaaS, PaaS services and provider services instances, https://azure.microsoft.com/en-us/pricing/details/cloud-services/, Mar. 2019.

[4] NBi Database Testing Framework, http://www.nbi.io/docs/installation-tools/, Apr. 2017.

[5] DbFit, Test-driven database development, http://dbfit.github.io/dbfit/, Mar. 2017.

[6] DbBench is a simple database benchmarking tool https://github.com/sj14/dbbench, Mar. 2017.

A Machine-Learning Framework for Supporting Intelligent Web-Phishing Detection and Analysis

Alfredo Cuzzocrea
University of Trieste
Trieste, Italy
alfredo.cuzzocrea@dia.units.it

Fabio Martinelli
Institute for Informatics and Telematics
Pisa, Italy
fabio.martinelli@iit.cnr.it

Francesco Mercaldo
Institute for Informatics and Telematics
Pisa, Italy
francesco.mercaldo@iit.cnr.it

ABSTRACT

This paper proposes a machine-learning framework for supporting intelligent web phishing detection and analysis, and provides its experimental evaluation. In particular we make use of state-of-the-art decision tree algorithms for detecting whether a Web site is able to perform phishing activities. If this is the case, the Web site is classified as a Web-phishing site. Our experimental evaluation confirms the benefits of applying machine learning methods to the well-known web-phishing detection problem.

CCS CONCEPTS

• **Security and privacy → Web application security**.

KEYWORDS

Web Phishing, Machine Learning for Supporting Web Phishing Detection, Web Phishing Analysis

ACM Reference Format:
Alfredo Cuzzocrea, Fabio Martinelli, and Francesco Mercaldo. 2019. A Machine-Learning Framework for Supporting Intelligent Web-Phishing Detection and Analysis. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3331076.3331087

1 INTRODUCTION

Web security is an emerging trend, especially in the novel big data context (e.g., [1, 6, 15]). Traditionally, Web security has been addressed by exploiting several approaches, such as *privacy-preserving methodologies* (e.g., [1]), *hidden Markov models* (e.g., [21]), *logic-based approaches* (e.g., [9]), and so forth.

This traditional challenge, which involves in both academic and industrial research issues, is now emerging again due to its tight relation with novel *big data trends* (e.g., [7, 10, 19]), which has originated some very interesting approaches, among which [8, 18] are noticeable ones.

Among several problems, *Web phishing* (e.g., [5, 16, 17, 20]) is of relevant interest at now. Phishing is a method to imitating a official websites or genuine websites of any organization such as banks, institutes social networking websites, etc. Mainly phishing is attempted to theft private credentials of users such as username, passwords, PIN number or any credit card details etc. Phishing is attempted by trained hackers or attackers. Phishing is mostly attempted by phishy e-mails. This kind of Phishy e-mails may contains phishy or duplicate link of websites which is generated by attacker. By clicking these kinds of links, it is redirected on malicious website and it is easily to theft your personal credentials. Phishing Detection is a technique to detecting a phishing activity. There are various methods proposed by so many researchers. Among them Data Mining techniques are one of the most promising technique to detect phishing activity. Data mining is a new solution to detecting phishing issue. So data mining is a new research trend towards the detecting and preventing phishing website.

Starting from these considerations and in order to overcome the performances obtained, according by current literature, in this paper we propose a machine learning based method able to identify whether a web page is able to perform phishing activities.

Figure 1 shows the big picture of our framework. As shown in Figure 1, in our reference application scenario, several *Web Users* are interaction with *Web Phishing Sites* (still unknown, of course), and the goal of our framework is just to detect the Web phishing sites and notify the users on. To this end, the component *Feature Extraction* is in charge of extracting suitable features to drive the machine-learning-based detection phase. Features are extracted and an ad-hoc *Built-In Dataset* is populated this way. Finally, the *Decision Tree Algorithms* run over the latter dataset and the Web phishing event notification is finally reported to the *Web Users.*

This paper extends the previous short paper [4], where we introduced the main ideas of the proposed framework.

2 DECISION-TREE ALGORITHMS FOR WEB PHISHING DETECTION

In this Section we describe the method we propose for web phishing attacks detection.

Table 1 shows the features considered in the following study.

In order to collect data, we consider the PhishTank dataset [1]: PhishTank is a free community site where anyone can submit, verify, track and share phishing data. This dataset is in the form of .csv file format.

The evaluation consists of two different stages: (i) we provide hypotheses testing, to verify whether the features vector exhibit different distributions for attacks and normal messages populations; and (ii) decision-tree machine learning analysis in order to assess

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06.
https://doi.org/10.1145/3331076.3331087

[1] https://archive.ics.uci.edu/ml/datasets/Website+Phishing

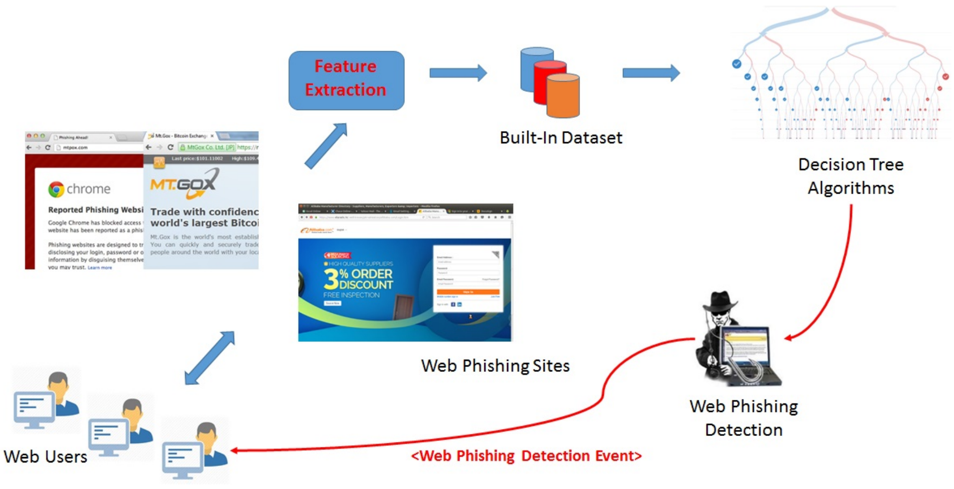


Figure 1: The Proposed Machine-Learning-Based Web Phishing Detection Framework

Variable	Feature
$F1$	URL Anchor
$F2$	Request URL
$F3$	Server Form Handler
$F4$	URL Length
$F5$	Having IP Address
$F6$	Prefix/Suffix
$F7$	IP
$F8$	Sub Domain
$F9$	Website Traffic
$F10$	Domain Age

Table 1: The Feature Set Involved in the Study

if the eight features are able to discriminate between attacks and normal messages.

Machine learning is a type of artificial intelligence able to provide computers with the ability to learn without being explicitly programmed [14].

Machine learning tasks are typically classified into two categories, depending on the nature of the learning available to a learning system:

- *Supervised learning*: the computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. It represents the classification: the process of building a model of classes from a set of records that contains class labels.
- *Unsupervised learning*: no labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

The algorithms considered are supervised decision tree-based i.e., they use a decision tree as a predictive model which maps observations about an item (represented in the branches) to conclusions about the target of the items value (represented in the leaves). These algorithms (i.e., *J48*, *HoeffdingTree*, *RandomForest*, *RetTree*, *LMT* and *DecisionStump*) are the most widespread to solve data mining problems [14] for instance, from malware detection [2, 3, 11, 13] to pathologies classification [12].

We consider in this work five different machine learning algorithms in order to enforce the conclusion validity. With regards to the hypotheses testing, the null hypothesis to be tested is:

H_0 : 'phishing and legitimate web pages exhibit similar values of the considered features'.

The null hypothesis was tested with Wald-Wolfowitz (with the p-level fixed to 0.05), Mann-Whitney (with the p-level fixed to 0.05) and with Kolmogorov-Smirnov Test (with the p-level fixed to 0.05). We chose to run three different tests in order to enforce the conclusion validity. The purpose of these tests is to determine the level of significance, i.e., the risk (the probability) that erroneous conclusions be drawn: in our case, we set the significance level equal to .05, which means that we accept to make mistakes 5 times out of 100. The analysis goal is to verify if the considered features are able to correctly discriminate between phishing and normal web pages. These algorithms were applied to the full feature vector. The classification analysis is performed using the Weka[2] tool, a suite of machine learning software, employed in data mining for scientific research.

3 EXPERIMENTAL ASSESSMENT AND ANALYSIS

We used five metrics in order to evaluate the results of the classification: Precision, Recall, F-Measure, MCC and RocArea. The results that we obtained with this procedure are shown in table 2.

[2] http://www.cs.waikato.ac.nz/ml/weka/

Algorithm	Precision	Recall	F-Measure	MCC	Roc Area	Class
J48	0,904	0,892	0,898	0,829	0,958	legitimate
	0,923	0,916	0,919	0,833	0,958	phishing
HoeffdingTree	0,840	0,892	0,865	0,770	0,948	legitimate
	0,882	0,916	0,899	0,786	0,953	phishing
RandomForest	0,891	0,892	0,892	0,818	0,968	legitimate
	0,917	0,912	0,914	0,822	0,966	phishing
RepTree	0,856	0,911	0,882	0,799	0,964	legitimate
	0,933	0,872	0,901	0,804	0,961	phishing
LMT	0,876	0,892	0,884	0,804	0,970	legitimate
	0,922	0,905	0,913	0,821	0,972	phishing
DecisionStump	0,794	0,849	0,820	0,692	0,835	legitimate
	0,836	0,913	0,873	0,726	0,845	phishing

Table 2: Classification results.

As shown in Table 1 the proposed method is able to obtain a precision equal to 0,923 and a recall equal to 0,916 in phishing attack detection using the J48 algorithm. The classification algorithms obtaining the best precision are J48 and RepTree, but considering also the recall metric, we highlight that the RepTree recall is lower if compared with the one obtained by the J48 classification algorithm: this is the reason why we confirm the J48 algorithm as the one obtaining the best performances in terms of precision and recall in order to detect web phishing attacks. As a matter of fact, the remaining algorithms (i.e., HoeffdingTree, RandomForest, LMT and DecisionStump) exhibit lower performances than J48 and RepTree in terms of precision and recall.

4 CONCLUSIONS AND FUTURE WORK

Recently, a more effective approach to fight phishing that relies on machine learning techniques has emerged. In this approach, models extracted by a ML technique are used to classify websites either as legitimate or phishy, based on certain features. In this paper we propose a method machine learning-based able to detect whether a web page exhibits phishing attacks. As future work, we plan to extend the proposed features in order to increase the detection accuracy, in addition we plan to apply formal methods with the aim to detect the code in which the malicious action happens.

ACKNOWLEDGMENTS

This work has been partially supported by H2020 EU-funded projects SPARTA contract 830892 and C3ISP and EIT-Digital Project HII.

REFERENCES

[1] Saad A. Abdelhameed, Sherin M. Moussa, and Mohamed E. Khalifa. 2018. Privacy-preserving tabular data publishing: A comprehensive evaluation from web to cloud. *Computers & Security* 72 (2018), 74–95.

[2] Pasquale Battista, Francesco Mercaldo, Vittoria Nardone, Antonella Santone, and Corrado Aaron Visaggio. 2016. Identification of Android Malware Families with Model Checking. In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy, ICISSP 2016, Rome, Italy, February 19-21, 2016.* SciTePress, 542–547.

[3] Gerardo Canfora, Francesco Mercaldo, Corrado Aaron Visaggio, and Paolo Di Notte. 2014. Metamorphic malware detection using code metrics. *Information Security Journal: A Global Perspective* 23, 3 (2014), 57–67.

[4] Alfredo Cuzzocrea, Fabio Martinelli, and Francesco Mercaldo. 2018. Applying Machine Learning Techniques to Detect and Analyze Web Phishing Attacks. In *iiWAS.* ACM, 355–359.

[5] Serge Egelman, Lorrie Faith Cranor, and Jason I. Hong. 2008. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, 2008, Florence, Italy, April 5-10, 2008.* 1065–1074.

[6] Hsiu-Chuan Huang, Zhi-Kai Zhang, Hao-Wen Cheng, and Shiuhpyng Winston Shieh. 2017. Web Application Security: Threats, Countermeasures, and Pitfalls. *IEEE Computer* 50, 6 (2017), 81–85.

[7] IBM, Paul Zikopoulos, and Chris Eaton. 2011. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data* (1st ed.). McGraw-Hill Osborne Media.

[8] Igor V. Kotenko, Igor Saenko, and Alexander Branitskiy. 2018. Applying Big Data Processing and Machine Learning Methods for Mobile Internet of Things Security Monitoring. *J. Internet Serv. Inf. Secur.* 8, 3 (2018), 54–63.

[9] Rafal Kozik, Michal Choras, and Witold Holubowicz. 2017. Packets tokenization methods for web layer cyber security. *Logic Journal of the IGPL* 25, 1 (2017), 103–113.

[10] Kuan-Ching Li, Hai Jiang, Laurence T. Yang, and Alfredo Cuzzocrea (Eds.). 2015. *Big Data - Algorithms, Analytics, and Applications.* Chapman and Hall/CRC.

[11] Fabio Martinelli, Fiammetta Marulli, and Francesco Mercaldo. 2017. Evaluating Convolutional Neural Network for Effective Mobile Malware Detection. *Procedia Computer Science* 112 (2017), 2372–2381.

[12] Francesco Mercaldo, Vittoria Nardone, and Antonella Santone. 2017. Diabetes Mellitus Affected Patients Classification and Diagnosis through Machine Learning Techniques. *Procedia Computer Science* 112, C (2017), 2519–2528.

[13] Francesco Mercaldo, Corrado Aaron Visaggio, Gerardo Canfora, and Aniello Cimitile. 2016. Mobile malware detection in the real world. In *Software Engineering Companion (ICSE-C), IEEE/ACM International Conference on.* IEEE, 744–746.

[14] Tom M Mitchell. 1999. Machine learning and data mining. *Commun. ACM* 42, 11 (1999), 30–36.

[15] Paulo Jorge Costa Nunes, Iberia Medeiros, José Fonseca, Nuno Neves, Miguel Correia, and Marco Vieira. 2018. Benchmarking Static Analysis Tools for Web Security. *IEEE Trans. Reliability* 67, 3 (2018), 1159–1175.

[16] Nuttapong Sanglerdsinlapachai and Arnon Rungsawang. 2010. Using Domain Top-page Similarity Feature in Machine Learning-Based Web Phishing Detection. In *WKDD.* IEEE Computer Society, 187–190.

[17] Guang Xiang, Jason I. Hong, Carolyn Penstein Rosé, and Lorrie Faith Cranor. 2011. CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites. *ACM Trans. Inf. Syst. Secur.* 14, 2 (2011), 21:1–21:28.

[18] Zheng Xu, Zhiguo Yan, Lin Mei, and Hui Zhang. 2015. The Big Data Analysis of the Next Generation Video Surveillance System for Public Security. In *WEB (Lecture Notes in Business Information Processing)*, Vol. 258. Springer, 171–175.

[19] Chao-Tung Yang, Jung-Chun Liu, Ching-Hsien Hsu, and Wei-Li Chou. 2014. On improvement of cloud virtual machine availability with virtualization fault tolerance mechanism. *The Journal of Supercomputing* 69, 3 (2014), 1103–1122.

[20] Yue Zhang, Jason I. Hong, and Lorrie Faith Cranor. 2007. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007.* 639–648.

[21] Zhongliu Zhuo, Yang Zhang, Zhi-Li Zhang, Xiaosong Zhang, and Jingzhong Zhang. 2018. Website Fingerprinting Attack on Anonymity Networks Based on Profile Hidden Markov Model. *IEEE Trans. Information Forensics and Security* 13, 5 (2018), 1081–1095.

sElect: Secure Election as a Service

Mohamed Nassar, Bassel Rawda, Mohamed Mardini
Computer Science Department
Faculty of Arts and Sciences
American University of Beirut (AUB)
[mn115|mjm26|brr01]@aub.edu.lb

ABSTRACT

Online voting is a challenging socio-technical problem that is still an open research question. Current approaches are based on involved cryptographic solutions that hardly can be explained to the average voter. Still, attackers may be able to circumvent the system without breaking the cryptographic protocols. In this paper, we demonstrate a toy voting protocol based on additive homomorphic encryption and two non-colluding parties. The protocol design aims at preserving the essential security properties such as election integrity and voter anonymity while being much simpler to explain. We propose a RESTful implementation of a web front-end of the proposed system. We highlight the fact that cryptography is only one facet of security. By conducting several web attack scenarios to test the robustness of the web interface, we show that the system may still be compromised without breaking the cryptography.

CCS CONCEPTS

• **Security and privacy → Privacy-preserving protocols**; **Security protocols**; *Web application security*; *Social aspects of security and privacy*;

KEYWORDS

Online Voting, Cybersecurity, Homomorphic Encryption, Paillier Encryption

ACM Reference Format:
Mohamed Nassar, Bassel Rawda, Mohamed Mardini. 2019. sElect: Secure Election as a Service. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3331076.3331109

1 INTRODUCTION

Voting is a method for a group of people to collectively elect a person, take a decision or express an opinion. Online voting systems are gaining acceptance with the widespread use of secure web services and cloud computing such as electronic currency and online banking. However, many researchers agree that online voting is hard. It has challenging privacy, security and accountability issues. We propose a cryptographic solution that is simple to explain to the voters. Our protocol is based on partially homomorphic encryption and two non-colluding parties. We provide a RESTful implementation of the voting framework using micro and distributed web services. More importantly, we recognize that the correctness of the cryptography does not ensure security, and that security is a process rather than just a product. Therefore, we pen-test our web services for the most known web and remote vulnerabilities.

2 RELATED WORK

Many secure and verifiable voting schemes are proposed with solid cryptographic foundations. They ensure the verifiability of the results (also known as the integrity property) and in the same time deny any match between the votes and the corresponding voters (also known as the ballot secrecy property). State of the art protocols use mix nets and homomorphic encryption. Many online systems offer verifiable elections such as Helios (https://vote.heliosvoting.org/). With the arising popularity of Blockchain applications, voting has also been formalized in terms of a smart contract [4]. However, online voting has many practical constraints. Helios 2.0 is shown to be vulnerable to a man-in-the-middle attack by installing a browser rootkit that detects the ballot web page and modifies votes [3]. Blockchain has its security weaknesses. Several attacks against smart contracts are surveyed in [1]. In [8] the experience of attacking the Washington, D.C. Internet voting system highlights its many weaknesses. Within 48 hours of the system going live, it was almost completely compromised. Many countries dropped electronic voting for absentee overseas voters over cybersecurity fears[1]. Researchers consider voting as hard and suggest physical redundancy such as tally papers to accompany any online system [2]. The reason is that online voting is not able to deal with compromises after they have occurred like in the case of online banking. In e-banking, transactions, statements, and logs allow customers to detect fraudulent transactions. The banking fraud is considered a marginal cost of doing business. Internet voting systems are not similar since they deny fine-grained logs that may compromise the identity of the voters.

3 VOTING FRAMEWORK AND PROTOCOL

The voting protocol is based on a previous work of a subset of the authors [6]. A front-end broker is introduced here as a trusted entity to manage the communication in between the voter and two back-end servers: the trustee server which distributes sealed voting envelopes, and the tally server which counts votes under encryption. At the end of the election. the trustee verifies the integrity using a cryptographic identity as shown in Fig. 1. We use Paillier's cryptosystem [7] for its ability to add votes under encryption. This cryptosystem is assymetric (public/private keys), probabilistic and

IDEAS'19, June 10–12, 2019, Athens, Greece
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6249-8/19/06.
https://doi.org/10.1145/3331076.3331109

[1] http://www.reuters.com/article/us-france-election-cyber-idUSKBN16D233

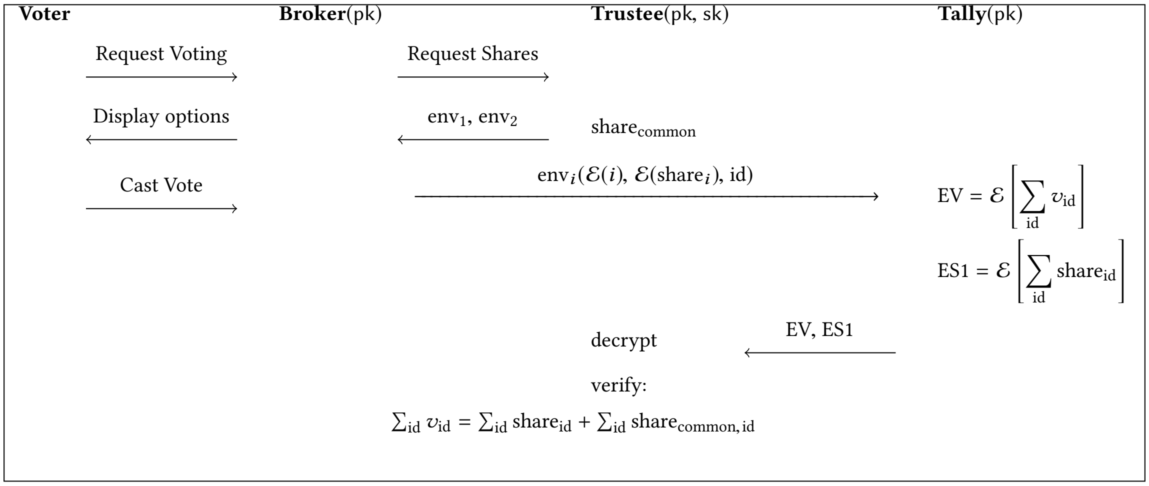


Figure 1: Voting protocol communication diagram

IND-CPA (Indistinguishable under chosen plain-text attack). A high performance implementation of this cryptosystem is proposed in [5]. The voting protocol allows binary votes (0 or 1). The voter solicits its two envelopes from the trustee server. The trustee decomposes each choice into two shares. One of the shares is common across the two choices:

$$0 = share_0 + share_{common}$$
$$1 = share_1 + share_{common}$$

The trustee encrypts the shares, gives the vote an id and generates two envelops:

$$env_1 = (\mathcal{E}_{\mathrm{pk}}(0) \mid\mid \mathcal{E}_{\mathrm{pk}}(share_0) \mid\mid id)$$
$$env_2 = (\mathcal{E}_{\mathrm{pk}}(1) \mid\mid \mathcal{E}_{\mathrm{pk}}(share_1) \mid\mid id)$$

The voter chooses only one of the two envelops, under an oblivious transfer protocol, and casts the chosen envelop to the tally. The tally keeps two separate counters: one for the votes and one for the shares. The two counters are transferred to the trustee at the end of the election for verification. More details, possible malleability attacks and countermeasures are discussed in [6].

4 SECURITY ASSESSMENT

It is clear from our implementation experience that it is not sufficient to have the cryptography correct in order to create a secure online voting system, not even close. The system can be vulnerable to many web threats such as:

- SQL injection. Tests were accomplished using SQLMap (http://sqlmap.org/)
- Cross-site scripting. Tests were accomplished using Xsser (https://tools.kali.org/web-applications/xsser)
- Brute force hash and password cracking attacks for weak admin passwords.
- Flooding and Denial of Service, we simulated stress conditions using Artillery (https://artillery.io/), a modern load testing toolkit.

The framework is further available for testing in form of a virtual machine that can be downloaded at https://drive.google.com/file/d/1Et7-6Wt1dUFbOjnkAtM_KP83qntU_U25/view?usp=sharing. The VM has a video with instructions to configure and run the voting web service.

5 CONCLUSION AND FUTURE WORK

Web developers may mistakenly estimate that implementing an election application is straightforward. However, security is a substantial issue that is critical to maintain fair and anonymous voting. Security in voting systems is challenging given the multitude of attack vectors and vulnerabilities. In this paper, we have proposed sElect, an implementation of a binary voting system using homomorphic encryption. We stressed on the fact that getting the cryptography right is not enough, and made the framework publicly available for further "black hat" testing. In future work, we will look at blockchain-based voting frameworks and assess their security.

REFERENCES

[1] N. Atzei, M. Bartoletti, and T. Cimoli. A survey of attacks on ethereum smart contracts (sok). In *Principles of Security and Trust*, pages 164–186. Springer, 2017.
[2] Bruce Schneier. Voting Security. IEEE Security & Privacy. https://www.schneier.com/essays/archives/2004/07/voting_security.html, July/August 2004. [Online; accessed 2015-07-24].
[3] S. Estehghari and Y. Desmedt. Exploiting the client vulnerabilities in internet e-voting systems: Hacking helios 2.0 as an example. *EVT/WOTE*, 10:1–9, 2010.
[4] P. McCorry, S. F. Shahandashti, and F. Hao. A smart contract for boardroom voting with maximum voter privacy. In *International Conference on Financial Cryptography and Data Security*, pages 357–375. Springer, 2017.
[5] M. Nassar, A. Erradi, and Q. M. Malluhi. Paillier's encryption: Implementation and cloud applications. In *2015 International Conference on Applied Research in Computer Science and Engineering (ICAR)*, pages 1–5. IEEE, 2015.
[6] M. Nassar, Q. Malluhi, and T. Khan. A scheme for three-way secure and verifiable e-voting. In *15th ACS/IEEE international conference on computer systems and applications (AICSSA'18)*, 2018.
[7] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.
[8] S. Wolchok, E. Wustrow, D. Isabel, and J. A. Halderman. Attacking the washington, dc internet voting system. In *Financial Cryptography and Data Security*, pages 114–128. Springer, 2012.

Blockchain-based Micropayment Systems: Economic Impact

Nida Khan
University of Luxembourg
Luxembourg
nida.khan@uni.lu

Tabrez Ahmad
ArcelorMittal Europe
Luxembourg
tabrez.ahmad@arcelormittal.com

Radu State
University of Luxembourg
Luxembourg
radu.state@uni.lu

ABSTRACT

The inception of blockchain catapulted the development of innovative use cases utilizing the trustless, decentralized environment, empowered by cryptocurrencies. The envisaged benefits of the technology includes the divisible nature of a cryptocurrency, that can facilitate payments in fractions of a cent, enabling micropayments through the blockchain. Micropayments are a critical tool to enable financial inclusion and to aid in global poverty alleviation. The paper conducts a study on the economic impact of blockchain-based micropayment systems, emphasizing their significance for socioeconomic benefit and financial inclusion. The paper also highlights the contribution of blockchain-based micropayments to the cybercrime economy, indicating the critical need of economic regulations to curtail the growing threat posed by the digital payment mechanism.

CCS CONCEPTS

• **Information systems → Digital cash**; • **Applied computing → Economics**; *Electronic commerce*; • **Social and professional topics → Financial crime**.

KEYWORDS

Blockchain, Cryptocurrency, Micropayments, Economic Impact, Cybercrime

ACM Reference Format:
Nida Khan, Tabrez Ahmad, and Radu State. 2019. Blockchain-based Micropayment Systems: Economic Impact. In *23rd International Database Engineering Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3331076.3331096

1 INTRODUCTION

Micropayments come in the category of electronic payment systems, which are financial transactions that take place through an electronic medium without using paper checks or cash. A micropayment is a financial transaction involving an amount of money less than a dollar or even a fraction of a cent but a definite number has not been agreed upon as a standard beyond which payment values fall into micropayments as seen in the nomenclature assigned in related literature [12, 14, 19]. High transaction fees is seen as a limiting factor for conducting micropayments by conventional payment solutions and practical implementations to bring about a seamless deployment of micropayments below a dollar, still remains an area of research. Information economy, dominated by the presence of digital goods like blog posts and digital services like online newspapers, has ushered a new era of payments, that involve very small amounts. Micropayments provide the tool to harness the economic benefits that can be reaped from such a market.

Blockchain is an immutable ledger of transactions, recorded in a decentralized distributed database and facilitates transactions in a trustless environment bringing down the costs associated with intermediaries in financial transactions [16]. However, scalability and performance issues are a bottleneck to optimum large-scale utilization of the blockchain technology [7]. This paper is a pioneer in analyzing the economic impact of blockchain-based micropayment systems. The paper gives the relevant background and related work in section 2. Economic impact of blockchain-based micropayments is discussed in section 3, while the implications for the cybercrime economy are elaborated in section 4. Conclusion is given in section 5.

2 BACKGROUND AND RELATED WORK

A micropayment provider can reduce the transaction fee to facilitate payments of small amounts. Apple launched the iTunes store in which songs are sold for 99 cents and Google Play also enabled micropayments as low as 10 cents per song. Both technology giants, Apple and Google, handle these micropayments by employing a probabilistic model for user behaviour to pick an optimal time to balance credit risk versus transaction fee by batching several consumer purchases into one. Cryptocurrencies are digital assets that are used for conducting payments in blockchain platforms and they can be used to develop micropayment systems that enable payments in fractions of a cent. The divisibility property of cryptocurrencies aids in conducting micropayments and a low cost/ nearly zero transaction fee model adopted by a blockchain platform can help to position it as a blockchain-based micropayment system. Lightning Network, Raiden and Stellar are few examples of blockchain-based micropayment systems. However, the use of these systems is presently limited on account of the technological issues and an absence of explicit regulations to govern the usage of cryptocurrencies [17], as in cases of loss of funds there is no redressal mechanism to compensate the users.

Chohan elaborates on the monetary role cryptocurrencies can play in hyperinflation [3]. Nica *et al.* discuss the economic benefits and risks of cryptocurrencies [17]. Fry and Cheah use econophysics models to examine shocks and crashes in cryptocurrency markets [9]. Li and Wang discuss the technology and economic determinants of the Bitcoin exchange rate [13]. Budish focuses on the economic limits of Bitcoin, indicating skepticism and caution about large-scale uses of the technology [2]. The present work deals with a study of the economic impact of blockchain-based micropayment

IDEAS'19, June 10–12, 2019, Athens, Greece

ACM ISBN 978-1-4503-6249-8/19/06.
https://doi.org/10.1145/3331076.3331096

systems, including the adverse contribution of such systems to cybercrimes.

3 ECONOMIC IMPACT

Blockchain-based micropayment systems employ the use of cryptocurrencies to facilitate payments and as such are susceptible to all risks and provide all benefits that come in the domain of cryptocurrencies. However, being facilitators of micropayments, some additional beneficial impacts are observed which can aid in the economic progress of the society. The usage of these systems in dark markets and cybercrime has been discussed in section 4. The blockchain platform being used also has an impact on the micropayment system's economic viability. Micropayment systems employing proof of work consensus incur high energy consumption costs and are economically limiting [2], making the platforms unsustainable for long term usage. However, the advent of blockchain platforms using other consensus mechanisms have provided a remedy to this undesirably high utilization of electricity, paving the way for economically viable blockchain-based micropayment systems.

3.1 Socioeconomic Benefit

The world can be divided into four income groups [23]. There are approximately 1 billion people in Level 1 that live on less than $2 a day, who do not have even the basic necessities of life. Level 2 contains 3 billion people surviving on $2 to $8 a day. Level 3 has 2 billion people, whose needs encompass around $8 to $32 per day. Level 4 has 1 billion people living on more than $32 a day [23]. When we think of payment systems, it is clear that the mainstream financial organizations and payment systems do not cater to people in Level 1 and Level 2, which together make more than half the present worlds' population. The rare who do cater like PayPal, have a fee structure which makes it infeasible to send amounts in fractions of a cent [18]. The people in Level 1 and Level 2 can certainly benefit from donations less than a dollar. Thus, from a socioeconomic perspective and to aid in the achieving of a few Sustainable Development Goals of United Nations Development Program [20], blockchain-based micropayment systems are a breakthrough in being able to transfer fractions of a cent, in real time and at extremely low cost, if not free.

3.2 Revenue Generation

The feasibility of micropayments brought about by blockchain-based systems can aid in further development of the market of microproducts [25], where products cost less than a few dollars. The development of e-commerce, virtual goods, online games, digital advertising, social networks and sale of digital information has revamped the need for low fee-based, instant payment transfers of small amounts. If no intermediaries are involved in the process, then this further helps to make this alternative market more independent. Stellar involves anchors but it's extremely low cost transactions and use of blockchain make it a much better technology for micropayments than existing micropayment systems. Global mobile app revenue in 2016 was $88 billion and it is expected to grow to $189 billion by 2020 [8]. App revenues are mostly generated by advertising and in-app purchases that involve micropayments. When Apple introduced the micropayment pricing model in 2009, then by 2017 around 50% of mobile app revenue was generated through in-app purchases involving amounts of the order of $0.99, $1.99 and $2.99. These new products and market can add to the revenue generation of an economy.

3.3 Elimination of Foreign Exchange and Enhancement of Stability

Cryptocurrencies like BTC, from Bitcoin blockchain platform, are seen as commodity money [5], which are deemed to be comparatively more transparent indicating any tampering done with it. Commodity money strengthens the social obligations of the issuer with respect to the wider society dependent on that currency [11] . Further commodity money has long been associated with price stability [6]. Consequently Forex traders have been observed to secure their funds in BTC during periods of volatility to hedge against the instability of fiat currency. Stellar's native cryptocurrency, XLM, is however inflationary [24] and is developed more to integrate the blockchain platform into the existing fiat system as opposed to Bitcoin and Ethereum, who seek to provide a more stable monetary system, where a commodity serves as the anchor to stabilize. In the absence of government regulations, cryptocurrencies have witnessed very high fluctuations being used mainly for speculation and investment. A sound regulatory environment and a well-designed cryptocurrency has the potential to be a global digital currency, eliminating the need for foreign exchange. Research is ongoing towards this end as seen in stablecoins [21].

3.4 Effect on Monetary Policy

Bitcoin was recognized to be private money by the German government [4]. In an economic system where private money issue is permitted, the nature of optimal monetary policy changes significantly [27]. Fiat money is inconvertible and cannot be redeemed whereas private money like Bitcoin can be redeemed in outside money. Even amidst frictions in the functioning of the private banking system, private money allows for the intermediation of investment whereas fiat currencies do not, making private money superior to fiat [26]. Besides private money has the property to be elastic and it's quantity can respond to shocks in a way that a stock of fiat currency cannot [27]. On account of the above reasons, it can be envisaged that with scalability and low transaction costs, private money like a cryptocurrency, has the possibility to be used in transactions involving goods, instead of fiat currency [25].

3.5 Financial Inclusion

The world presently has around 1.7 billion financially excluded adults. According to statistics provided by FINCA International, 76% of the poorest people, in 20 countries across Africa, Eurasia, Latin America, the Middle East and South Asia, are financially excluded [10]. Blockchain-based micropayment systems can provide to people in the lower income group, Level 1 and Level 2, to have access to a means of payment for buying microproducts as well as to store, send and receive payments of small amounts in their community. It can prove to be an alternative for banking services for the lower strata of society. An increasingly high number of people are relying on e-payments in the world. However, the value of a card payment, in nominal terms, has declined over the last decade and a half, from over $60 to less than $40. The smallest average value of a card payment was around $8 in 2016 [1] but card-based payments have no well-defined standard to cater to micropayments, leaving an unexploited avenue. Blockchain-based micropayment systems can fill this gap while accelerating the process of financial inclusion.

4 CONTRIBUTION TO THE CYBERCRIME ECONOMY

The economic definition of money implies usage of a currency as a unit of account, a medium of exchange and a store of value [17]. However, cryptocurrencies have been majorly used as a store of value, serving as assets for speculative purposes and as investment instruments. Their usage as a medium of exchange has been observed majorly in dark markets and online vendors selling illegal goods [17]. Blockchain-based micropayment systems use cryptocurrencies as a medium of exchange and pose the threat of being used as enumerated above with the additional risk of being used in microlaundering. Cybercrime economy is generating a minimum revenue of $1.5 trillion per annum with illicit and illegal online markets contributing to approximately $860 billion annually to the revenue [15]. Digital payments like PayPal are being used to engage in microlaundering techniques [22]. So far, cryptocurrencies account for only 4% of money laundered, which is equivalent to $80 billion per year [15], but they remain a potential medium for further growth of this economy. Blockchain-based micropayment systems like Lightning Network and Raiden, which provide the facility of conducting multiple micropayments without the payment transactions being recorded on the main blockchain platform can only serve to further this economy in the future. Cybercriminals reinvest the money in illegal trafficking of drugs, terrorist activities and further cybercrime. Economic regulations and adequate Anti-Money laundering/ Combating the Financing of Terrorism (AML/ CFT) measures need to be implemented to target this growing threat posed by all digital payments.

5 CONCLUSION

In this paper, we conducted a study on the economic impact of blockchain-based micropayment systems and highlighted the contribution of such systems to the cybercrime economy. An analysis of the economic impact indicates that the low cost micropayment model provided by blockchains can serve to reach the underbanked, expanding the cryptocurrency user base. These systems can aid in poverty alleviation by facilitating donations of a few dollars, with the blockchain ensuring that the funds reach the intended recipients. Blockchain-based micropayment systems can promote the development and increase the revenue stream, from the microproducts market. The absence of regulations has made cryptocurrencies vulnerable for exploitation in illegitimate uses. A study of the contribution of such digital payment mechanisms to the cybercrime economy brings out the critical need for the formulation and implementation of economic regulations and preventive laws, to intercept and disrupt cybercrime.

REFERENCES

[1] Morten Linnemann Bech, Umar Faruqui, Frederik Ougaard, and Cristina Picillo. 2018. Payments are a-changin' but cash still rules. *BIS Quarterly Review* (2018). https://www.bis.org/publ/qtrpdf/r_qt1803g.htm

[2] Eric Budish. 2018. The Economic Limits of Bitcoin and the Blockchain. *National Bureau of Economic Research* (2018). http://www.nber.org/papers/w24717

[3] Usman W. Chohan. 2019. Cryptocurrencies and Hyperinflation. *Notes on the 21st Century (CBRI)* (2019). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3320702

[4] Matt Clinch. 2013. Bitcoin recognized by Germany as 'private money'. Retrieved May 11, 2019 from https://www.cnbc.com/id/100971898

[5] CNBC. 2018. Virtual currencies are commodities, US judge rules. Retrieved May 10, 2019 from https://www.cnbc.com/2018/03/07/cryptocurrencies-like-bitcoin-are-commodities-us-judge-rules.html

[6] Bordo Michael D., Dittmar Robert D, and Gavin William T. 2003. Gold, Fiat Money, and Price Stability. *National Bureau of Economic Research* (2003).

[7] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. 2017. BLOCKBENCH: A Framework for Analyzing Private Blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17)*. ACM, New York, NY, USA, 1085–1100. https://doi.org/10.1145/3035918.3064033

[8] Artyom Dogtiev. 2018. App Revenues (2017). Retrieved May 10, 2019 from http://www.businessofapps.com/data/app-revenues/

[9] John Fry and Eng-Tuck Cheah. 2016. Negative bubbles and shocks in cryptocurrency markets. *Elsevier-International Review of Financial Analysis* 47 (2016), 343–352. https://doi.org/10.1016/j.irfa.2016.02.008

[10] Scott Graham. 2018. The 2017 Global Findex: A Fresh Look at Reaching the Unbanked. Retrieved May 11, 2019 from https://finca.org/blogs/2017-global-findex-a-fresh-look/

[11] Kenneth Hermele. 2014. Commodity Currencies vs Fiat Money: Automaticity vs Embedment. *FESSUD-Financialisation, Economy, Society and Sustainable Development* 37, 44 (Working Paper Series) (2014).

[12] Henrik Kniberg. 2002. *What Makes a Micropayment Solution Succeed*. Master's thesis. KTH Institution for Applied Information Technology.

[13] Xin Li and Chong Alex Wang. 2017. The technology and economic determinants of cryptocurrency exchange rates: The case of Bitcoin. *Elsevier-Decision Support Systems* 95 (2017), 49–60. https://doi.org/10.1016/j.dss.2016.12.001

[14] M.S. Manasse. 1995. The Millicent protocols for electronic commerce. In *Proceedings of the First USENIX Workshop on Electronic Commerce, July 11 - 12, 1995, New York, New York, USA*. 117–123.

[15] Michael McGuire. 2018. Understanding the Growth of the Cybercrime Economy *(RSA Conference)*.

[16] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved May 8, 2019 from https://bitcoin.org/bitcoin.pdf

[17] Octavian Nica, Karolina Piotrowska, and Klaus Reiner Schenk-Hoppe. 2017. Cryptocurrencies: Economic Benefits and Risks. *University of Manchester, FinTech working paper no. 2* (2017). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3059856

[18] PayPal. 2019. Buying something with PayPal. Retrieved May 2, 2019 from https://www.paypal.com/uk/webapps/mpp/paypal-fees

[19] PayPal. 2019. What are micropayments ? Retrieved May 8, 2019 from https://www.paypal.com/us/smarthelp/article/what-are-micropayments-faq664

[20] United Nations Development Programme. 2012. Sustainable Development Goals. Retrieved May 10, 2019 from http://www.undp.org/content/undp/en/home/sustainable-development-goals.html

[21] BitMEX Research. 2018. A brief history of Stablecoins (Part 1). Retrieved May 10, 2019 from https://blog.bitmex.com/a-brief-history-of-stablecoins-part-1/

[22] Jean-Loup Richet. 2013. Laundering Money Online: a review of cybercriminals methods. *CoRR* abs/1310.2368 (2013). arXiv:1310.2368 http://arxiv.org/abs/1310.2368

[23] Hans Rosling, Anna Rosling Ronnlund, and Ola Rosling. 2018. *Factfulness: Ten Reasons We're Wrong About the World–and Why Things Are Better Than You Think*. Flatiron Books, New York.

[24] Stellar. 2019. Inflation. Retrieved May 9, 2019 from https://www.stellar.org/developers/guides/concepts/inflation.html

[25] Ruy Alberto Valdes-Benavides and Paula Lourdes Hernandez-Verme. 2014. Virtual Currencies, Micropayments and Monetary Policy: Where Are We Coming from and Where Does the Industry Stand? *Journal of Virtual Worlds Research* 7, 3 (2014). https://doi.org/10.4101/jvwr.v7i3.7064

[26] Stephen D. Williamson. 1998. Private Money. In *Role of Central Banks in Money and Payments Systems (Federal Reserve Bank of Cleveland Conference)*. 469–499.

[27] Stephen D. Williamson. 2004. Limited participation, private money, and credit in a spatial model of money. *Economic Theory* 24, 4 (01 Nov 2004), 857–875. https://doi.org/10.1007/s00199-003-0463-3

A Genetic Algorithm for Discriminative Graph Pattern Mining

KAREL VACULIK, KD Lab FI MU, Czechia
LUBOS POPELINSKY, KD Lab FI MU, Czechia

CCS Concepts: • **Information systems** → **Data mining**; • **Mathematics of computing** → *Graph algorithms*; • **Computing methodologies** → *Supervised learning*.

ACM Reference Format:
Karel Vaculik and Lubos Popelinsky. 2019. A Genetic Algorithm for Discriminative Graph Pattern Mining. In *23rd International Database Engineering & Applications Symposium (IDEAS'19), June 10–12, 2019, Athens, Greece.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3331076.3331113

1 INTRODUCTION

In this paper, we introduce a new algorithm EWALDIS (Evolution- and random WALk-based algorihm for DIScriminative patterns) for mining discriminative patterns on the local level of dynamic attributed multigraphs. It uses a random walk-based approach [1] and a genetic algorithm to mine inexact patterns from the perspective of attributes and also timestamps. This also means that it does not require the discretization of the timestamps to be able to find some patterns. Moreover, by utilizing sampling techniques, the algorithm does not have to traverse the whole search space. EWALDIS is an improved version of WALDIS [7].

2 EXPERIMENTS

For experiments, we set the probability of edge non-selection to 0.05 and we performed 1000 random walks on each dataset. To assess EWALDIS, we trained and evaluated a classification model on a data created from discriminative patterns. We used k-NN classifier with $k = 3$ as our model. Given a train and a test set of events, both of size N, we first discovered patterns by the following procedure. We repeatedly selected n events from the train set at random and ran EWALDIS on these events to get several patterns. Then we assessed the existence of such patterns on both the training and the test set.

The assessment proceeds as follows for a given pattern and a given event:

(1) Select one graph from the pattern;
(2) Perform several random *walks* (10 by default) without restarts on this pattern graph while walking simultaneously in the tested instance.
(3) The score of this pattern graph is given by the sum of similarities of the simultaneously-walked edges divided by the number of all random-walk steps.
(4) If the random walk cannot continue in the tested instance, it continues only in the pattern graph and keeps counting the walks without similarities ;
(5) Pattern graph gets then the highest score across all random walks. Such a score is computed for each pattern graph and the average is returned as the final matching score.

By using this procedure, we compute the matching score for both positive and negative events from both training and test set. Then we use these matching scores from training data as new features and learned k-NN model on these data. The model is then evaluated on a test set created from matching scores obtained on the original test set.

We also created a simple baseline method for comparison with EWALDIS. The same classification algorithm was used, but the dataset features were prepared differently. Specifically, we created features from the edges adjacent to events. Each feature denotes an edge encoded by a pair (*label*, *relative_time-stamp*). For each event, the features had value either 0 or 1 depending on whether there was such an edge adjacent to the event.

3 RESULTS

We have evaluated the algorithm on real-world graph data like DBLP and Enron [1] We show in Table 1 that the method outperforms baseline algorithm for all data sets and that the increase of accuracy is quite high, between 2.5% for NIPS vs. KDD from DBLP dataset and 30% for Enron dataset. A C++ implementation and the datasets as well as the full version of this paper, are available at https://github.com/karelvaculik/ewaldis.

4 RELATED WORK

Methods for discriminative pattern mining generally assume two sets of instances: positive and negative and the goal is to find patterns with regard to a defined discriminative score. Existing work typically focuses on sets of graphs, i.e. one instance is represented as a graph and the pattern is its subgraph. Patterns are then used mostly for classification of input graphs. The task is in some extent different from ours as our positive and negative sets consist of vertex or edge events, and we search for patterns in the neighborhoods of these events.

Here we only bring a list of works [2, 5], MINDS [4], TGMiner [9], Waddling Random Walk algorithm [1], and a predictive pattern miner [3]. More extensive overview of related work can be found in the previous paper [7].

Settings			**Results accuracy**			
Data	**Pos.**	**Neg.**	**Baseline**		**EWALDIS**	
			train	**test**	**train**	**test**
DBLP	ICML	KDD	80.0	67.5	84.5	83.0
DBLP	KDD	ICML	74.5	64.0	80.5	79.0
DBLP	NIPS	KDD	80.0	79.0	86.0	81.5
DBLP	NIPS	ICML	63.0	53.0	74.5	65.0
ENR.	Bankr.	Bus.	87.5	57.5	95.0	90.0

Table 1. Experiment results

As a part of our research we also created a simpler algorithm WalDis [7] that is based on a simple greedy approach and does not use a genetic algorithm. Patterns found by this algorithm are simpler and may not capture complexities captured by patterns found by EWALDIS.

REFERENCES

[1] Han, G., and Sethu, H.: Waddling Random Walk: Fast and Accurate Mining of Motif Statistics in Large Graphs. In *ICDM*, pp 181–190, 2016.

[2] Kabutoya, Y., et al.: Dynamic Network Motifs: Evolutionary Patterns of Substructures in Complex Networks. In *Proceedings of the APWeb'11*, pp. 321–326, 2011.

[3] Nakagawa, K., et al.: Safe Pattern Pruning: An Efficient Approach for Predictive Pattern Mining. In *Proceedings of the 22nd ACM SIGKDD*, pp. 1785–1794, 2016.

[4] Ranu, S., et al.: Mining discriminative subgraphs from global-state networks. In *Proceedings of the 19th ACM SIGKDD*, pp. 509–517, 2013.

[5] Shen, E., and Yu, T.: Mining frequent graph patterns with differential privacy. In *Proceedings of the 19th ACM SIGKDD*, 545–553, 2013.

[6] Vaculík, K., and Popelínský, L.: DGRMiner: Anomaly Detection and Explanation in Dynamic Graphs. In *IDA*, Stockholm, Sweden, LNCS Springer, pp. 308–319, 2016.

[7] Vaculík, K., and Popelínský, L.: WalDis: Mining Discriminative Patterns within Dynamic Graphs. In IDEAS '18 Proceedings of the 22nd International Database Engineering & Applications Symposium. Calabria. ACM New York, 2018. s. 95-102.

[8] Wackersreuther, B., et al.: Frequent subgraph discovery in dynamic networks. In *MLG '10*. ACM, New York, NY, USA, pp. 155–162, 2010.

[9] Zong, B., et al.: Behavior Query Discovery in System-Generated Temporal Graphs. In *Proc. VLDB Endow.*, pp. 240–251, 2015.

[1] http://dblp.uni-trier.de/ http://www.cis.jhu.edu/~parky/Enron

Author List

Author List(Continued)

Author List(Continued)

The 23rd International Database
&
Applications Engineering Symposium

ISBN 978-1-4503-6249-8
10000 >
9 781450 362498